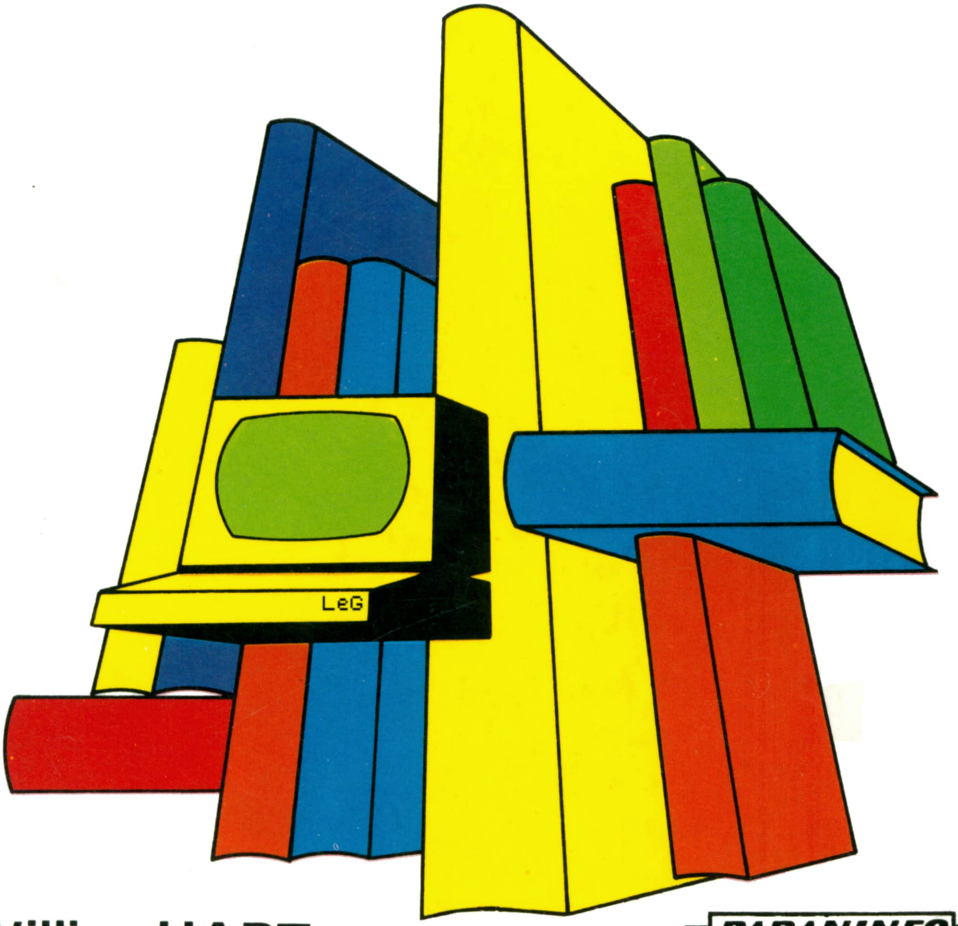
DICCIONARIO del BASIC



Willie HART

PARANINFO S.A.

Diccionario del BASIC

W. A. Hart

Diccionario del BASIC

SEGUNDA EDICION

1985

PARANINFO S.A.

MADRID

Traducido por:
FELISA MATEO GARCIA

© W. A. HART
© de la edición española Editorial Paraninfo, S. A., Madrid (España)
© de la traducción española Editorial Paraninfo, S. A., Madrid (España)

Título original
HART'S DICTIONARY OF BASIC

Published by:
Sigma Technical Press, Cheshire SK9 5DY, England

Distributed by:
John Wiley and Sons Ltd. Sussex P019 1UD, England

Reservados los derechos para todos los países. Ninguna parte de esta publicación, incluido el diseño de la cubierta, puede ser reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste electrónico, químico, mecánico, electro-óptico, grabación, fotocopia o cualquier otro, sin la previa autorización escrita por parte de la Editorial.

IMPRESO EN ESPAÑA
PRINTED IN SPAIN

ISBN: 0 905 104 27 7 (edición inglesa)
ISBN: 84-283-1378-4 (edición española)

Depósito Legal: M-25.292-1985

PARANINFO S.A. Magallanes, 25 - 28015 MADRID (3-3569)

ALCO, artes gráficas. Jaspe, 34 - 28026 - MADRID

Indice de materias

1. Introducción 7

2. BASIC y BASIC estándar 9

3. Definición de términos 11

3.1. Puntuación en las definiciones 11
3.2. Formato de instrucciones 11

4. Sugerencias para la traducción de programas BASIC 20

5. Visión general de Códigos para Micros 26

5.1.. Ejemplos sobre PEEK y POKE 28
5.2. Mapas de memoria y conversión hexadecimal/decimal . 33
5.3. Códigos para el color 43
5.4. Códigos para el sonido 45
5.5. Comparación entre los juegos de caracteres de los micros 46

6. Comentarios de los lectores 55

7. Relación alfabética de comandos, sentencias, funciones y operaciones del BASIC 57

1. Introducción

Antes de recopilar este Diccionario tenía el proyecto de hacer algo bastante distinto. Mi intención original era la de escribir un programa BASIC que efectuase la traducción de cualquier versión de BASIC a cualquier otra. Sin embargo, enseguida me encontré con que el BASIC no era tan "básico", sino más bien complejo; estaría mejor descrito si se le llamase NOBASIC. Por ejemplo, de las primeras 500 instrucciones soportadas en docenas de ordenadores, sólo un puñado son totalmente compatibles en todos ellos. También me encontré con que la traducción de palabra por palabra daba lugar a un código ineficaz y que no se beneficiaba de las características especiales proporcionadas por el hardware, que normalmente son las razones que en primer lugar le inducen a uno a comprar un ordenador en particular.

El objetivo principal de este Diccionario es ayudar a convertir programas BASIC escritos en una máquina para que funcionen en otra. El Diccionario no intenta ser un tutor o sustituir los libros de texto y los manuales de los fabricantes de ordenadores. Por el contrario, el Diccionario tratará de complementarlos, por lo tanto, Vd. deberá familiarizarse primero con el ordenador y con su versión del BASIC si desea una utilización óptima del Diccionario.

Así pues, el Diccionario tiene dos funciones primordiales. En primer lugar es un libro de referencia. La mayoría de los libros de texto sobre programación están estructurados para ayudar a la enseñanza, con las sentencias del lenguaje agrupadas por tipos. El Diccionario es muy sencillo tanto en el formato como en el estilo del lenguaje y está organizado alfabéticamente para facilitar las referencias.

En segundo lugar el Diccionario puede utilizarse como un cuaderno de notas. Se ha reservado un área de Notas de Traducción en cada página, de manera que Vd. puede anotar aquellas futuras referencias de traducciones (palabra por palabra o sus propias rutinas) que correspondan a su ordenador. El área de Notas también puede utilizarse para documentar instrucciones nuevas (y correcciones de los errores), a medida que vayan siendo anunciadas por los fabricantes. De hecho, hay unas cuantas instrucciones que todavía no tienen

traducción; se estaban investigando cuando este Diccionario se estaba imprimiendo.

Para la preparación de este Diccionario he recibido una extraordinaria ayuda y una entusiasta colaboración por parte de mucha gente. Quiero dar las gracias a los fabricantes y suministradores de los sistemas de ordenadores que he revisado, especialmente a APPLE, BBC-ACORN, PET, SHARP, SINCLAIR, TANDY y VIC; sin su ayuda habría sido necesario un esfuerzo mucho mayor por mi parte y le hubiese tenido que dedicar mucho más tiempo. También le debo mucho a Graham Beech y Stuart Johnston, por su generosa ayuda, su paciencia y comprensión mientras yo aprendía los misterios del negocio de los libros.

Permítanme recordar a los "oscuros compañeros" –Katrina y Linda– cuya habilidad para mecanografiar y organizar me evidenciaron el desorden de mis guiones.

Finalmente, pero no en último lugar, quiero mostrar mi agradecimiento a mi familia, Bárbara, Brian y Fiona, por los fines de semana perdidos, las llegadas tarde por las noches, las interminables tazas de té...

Gracias a todos y cada uno.

2. BASIC y BASIC estándar

El lenguaje de programación BASIC (Beginners All-purpose Symbolic Instruction Code), es el lenguaje más ampliamente aceptado para microordenadores; su uso también se ha extendido en minis y en grandes ordenadores.

El BASIC ha contribuido al crecimiento de la industria de los microordenadores ya que es relativamente fácil de aprender y los programas se pueden escribir con bastante rapidez. Sin embargo, la demanda de microordenadores y su creciente sofisticación han sobrepasado nuestra capacidad para producir software a un ritmo suficientemente rápido. Añada a esto el hecho de que los costes del software han aumentado espectacularmente y de que la gran mayoría de los microordenadores está en manos de usuarios relativamente inexpertos; entonces empezamos a ver la magnitud del problema. Una forma de resolver la situación, consistiría en hacer un mejor uso del software disponible en lugar de volver a inventar la rueda con cada tipo de ordenador. Sin embargo, una gran parte de este software se ha escrito en diferentes –y en cierto grado incompatibles– versiones de BASIC, siendo difícil de transportar de un ordenador a otro.

La falta de estandarización de las características software y hardware no es un problema nuevo. La última ocasión en que los grandes compradores y usuarios de ordenadores se reunieron y forzaron una estandarización dio por resultado el COBOL (COmmon Bussiness Oriented Language). Los programas COBOL requieren unos cambios mínimos para poder ser transferidos de un ordenador a otro, siempre y cuando, claro está, los programas se hayan escrito siguiendo las normas estándares del COBOL y del proceso de datos.

Por una variada gama de razones, no es factible la misma solución en un futuro inmediato. Por una parte, no hay usuarios potentes; por otra han aparecido muchos más fabricantes (grandes, medianos, pequeños y minúsculos); y finalmente existe una enorme cantidad de software desarrollado que se quedaría obsoleto si se cambiase el BASIC, aunque sólo fuera para una ligera ampliación.

Con el paso del tiempo este problema se resolverá: los microordenadores tendrán memorias mayores, los fabricantes suministrarán características más potentes y los compiladores e intérpretes BASIC se optimizarán para suministrar un código máquina mejor y más flexible y para permitir un lenguaje mucho más amplio.

Mientras tanto, tenemos el problema de la conversión de programas BASIC. Sinceramente espero que este libro le proporcione una solución.

3. Definición de términos

3.1. PUNTUACION EN LAS DEFINICIONES DE SENTENCIAS

El uso de puntuación (caracteres del tipo coma o punto y coma) y de declarativas de caracteres especiales (número de fichero, $-cadenas, paréntesis) no es consistente en las distintas formas del BASIC; en algunos casos la puntuación es opcional, en otros es obligatoria y en otros incluso contradictoria, ya que el significado es diferente del generalmente aceptado por la mayoría de versiones del BASIC. Por estas razones la puntuación se ha contemplado al mínimo en el texto de definición de las sentencias del Diccionario y normalmente, aunque no exclusivamente, sólo se ha utilizado cuando es obligatoria.

3.2. FORMATO DE INSTRUCCIONES

El formato de las instrucciones suele ser coincidente para las distintas versiones del BASIC. En algunos casos, aunque la escritura y el significado de un comando sean iguales, los parámetros que lo controlan son diferentes en cuanto al contexto y al significado. En particular, las instrucciones del tipo gráficos (COLOR, SOUND, DRAW, POSITION) están cualificadas por enteros que hacen referencia a tablas definidas en cualquier parte de un programa; el tamaño de las tablas y el significado de cada elemento es en principio diferente de una máquina a otra. Por razones de economía no se han incluido en este Diccionario todas estas posibilidades, aunque en la mayoría de los casos es posible establecer una traducción convencional.

Los formatos de instrucción utilizados en este Diccionario se ajustan a los estándares generalmente aceptados del BASIC. Para las abreviaturas se ha establecido una referencia a la traducción principal.

Se ha suprimido la duplicidad de expresiones en aquellos casos en que el significado de instrucciones similares dependa de un conjunto diferente de parámetros cualificadores.

Se incluyen ejemplos en una sección especial para facilitar la comprensión de los formatos de instrucciones. De nuevo por razones de economía, sólo se han incluido las instrucciones más complicadas.

Los distintos elementos de las instrucciones BASIC son los siguientes:

Mandato/sentencia (modo inmediato)

Un mandato, o una sentencia en modo inmediato, indica al intérprete del BASIC que debe ejecutar una tarea inmediatamente. Un comando no tiene número de línea y se introduce normalmente desde el teclado. Los comandos se escriben a continuación del carácter de diálogo (prompt) sin espacios intermedios.

Sentencia (modo diferido)

Una sententencia en modo diferido indica al intérprete del BASIC que debe ejecutar una tarea sólo cuando se ejecuta un programa completo. Una sentencia tiene un único número de línea (del 1 al 15999) siendo el Intérprete del Basic el encargado de mantenerlas ordenadas permitiendo así la introducción de las sentencias desde el teclado de forma totalmente aleatoria.

Función

Una función de nombre a una operación que se ejecutá utilizando uno o más valores de parámetros para devolver un solo valor como resultado. Las funciones numéricas (NUMERIC) estándares están identificadas por una palabra reservada de tres letras seguida de uno o más parámetros que van entre paréntesis y separados por comas.

Las funciones de tipo cadena (STRING) se identifican mediante palabras reservadas y sirven para manipular datos en forma de cadenas.

Las funciones del usuario (USER) están definidas por DEF FN y un nombre de función, y se las llama con FN y el nombre de función.

Operador

Un operador ejecuta una operación lógica o matemática a partir de uno o más valores, dando como resultado un único valor.

Ejemplos de operadores aritméticos:

+	SUMA O POSITIVO UNARIO
–	RESTA O NEGATIVO UNARIO
*	MULTIPLICACION
/	DIVISION
**	EXPONENCIACION (POTENCIA)
MOO	MODULO (RESTO ENTERO)

Ejemplos de operadores relacionales:

=	IGUAL	<=	MENOR QUE O IGUAL A
<	MENOR QUE	>=	MAYOR QUE O IGUAL A
>	MAYOR QUE	<>	NO IGUAL A

Los operaciones relacionales devuelven el valor 1 si la expresión es verdadera y 0 si no lo es.

Ejemplos de operadores lógicos:

AND	"Y" LOGICO
OR	"O" LOGICO
NOT	COMPLEMENTO LOGICO

Los operadores lógicos devuelven el valor 1 si la expresión es verdadera, y 0 si es falsa.

Otros operadores

COMA (COMMA)
causa la impresión por zonas
(de 4 a 8 zonas por línea dependiendo de la anchura de la pantalla)

PUNTO (PERIOD)
se utiliza para indicar el punto decimal

; PUNTO Y COMA (SEMI-COLON)
provoca la impresión continua, sin espacios ni zonas

: DOS PUNTOS (COLON)
permite múltiples sentencias por cada línea de programa

( ) PARENTESIS (PARENTHESES)
utilizados en expresiones para determinar el orden en el que se deben efectuar las operaciones matemáticas

@ A DE ARROBAS (AT SIGN)
hace que la impresión de la pantalla comience en un punto específico

SIGNO SOSTENIDO (NUMBER SIGN)
significa doble precisión en variables numéricas y arrays; indica un canal o un número de referencia de fichero

$ SIGNO DOLAR (DOLLAR SIGN)
significa que una variable o un array es de tipo cadena

! EXCLAMACION (EXCLAMATION)
significa simple precisión en variables/arrays numéricos

% TANTO POR CIENTO (PERCENTAGE)
significa que una variable o array es de tipo entero

? SIGNO DE INTERROGACION (QUESTION MARK)
abreviatura para el MANDATO PRINT

\ BARRA INCILINADA (BACK SLASH)
ver DOS PUNTOS

' APOSTROFO (APOSTROPHE)
abreviatura para el comando REM

& Y (AMPERSAND)
concatena cadenas.

Elementos de datos

Se pueden usar caracteres alfanuméricos y especiales para identificar un elemento de datos, y también para definir el tipo de datos que se pueden almacenar en un elemento determinado. Generalmente, una vez que se ha asignado un nombre, no puede ser utilizado de nuevo en el mismo programa. El convenio más ampliamente aceptado para la nomenclatura, consiste en una letra (A-Z), o una letra seguida de un dígito (O-9) para variables reales; las variables enteras y las cadenas tienen un tercer carácter (% y $ respectivamente); los

arrays o tablas suelen ser de un único carácter alfabético para los numéricos y con un $ para los arrays de cadenas aunque algunas veces está permitido poner dos letras y por lo menos en un caso estas letras duplicadas son necesarias.

En algunos sistemas se permiten los nombres de datos de tres o más caracteres (sin contar el $, %) y aunque se permitan más de dos caracteres, sólo intervienen los dos primeros (DOS y DOSCAR serían nombres duplicados). También se puede DECLARAR (DECLARE), al comienzo del programa, que todas las variables que empiecen por una letra específica son de tipo específico.

Variable

Una variable es un item de almacenamiento de datos cuyo valor puede cambiar durante la ejecución del programa. Los datos que se pueden albergar en una variable y la utilización tanto de la variable como de su contenido depende de su definición, pudiendo ser de tipo "numérico" o "cadena".

Los datos de una variable NUMERICA (NUMERIC) tienen dos tipos de formato:

ENTERO (INTEGER) que es un número con o sin signo escrito, sin el punto (o coma) decimal.

REAL (REAL) es una cantidad con o sin signo consistente en parte entera y parte decimal que incluye el punto decimal; el rango de valores cubierto por los números REALES depende del tipo de datos que puede ser "fijo" (número de decimales fijo) o "de punto flotante" (precisión doble o simple).

Debido al número de dígitos involucrados, los números en punto flotante se imprimen normalmente con notación científica. En el siguiente ejemplo, el número que hay detrás de la E extiende el número que hay delante de la E mediante la potencia correspondiente de 10.

4E5 = 4 X 10 (elevado a la potencia 5) = 400000
4E-5 = 4/10 (elevado a la potencia 5) = 0,00004

El rango de valores de los datos numéricos es:

Decimal punto fijo y Enteros	+/–32767
Punto flotante simple precisión	10 (elevado a +/–38)
Punto flotante doble precisión	10 (elevado a +/–127)

Una variable de tipo CADENA (STRING) es una secuencia de caracteres (máximo 256) que debe tratarse como una unidad; se permite cualquier combinación de caracteres ASCII; los valores asignados a las variables de tipo cadena se escriben entre comillas, el tamaño estándar, por defecto, para la longitud de una cadena suele ser de 10 ó 18 caracteres siempre que no venga definida por una sentencia DIM (en algunos BASIC esta sentencia es obligatoria).

Una variable SUBCADENA (SUBSTRING) es un número de caracteres consecutivos que se encuentran en una cadena y que pueden ser referenciados a través de unos subíndices o mediante funciones como MID$ y LEFT$. Si la cadena A$ contiene "W HART" entonces A$ (1,3) = MID$ (A$,1,3) = "W H".

Tabla (Array)

Un array consiste en un grupo de variables de idéntica naturaleza; cada una de las variables está referenciada por un nombre de array y un subíndice (una dimensión) o por dos subíndices de fila/columna (dos dimensiones); algunos sistemas permiten 3 dimensiones o más. Si la sentencia DIM no especifica otra cosa, los valores tomados por defecto para los arrays son de 11 caracteres (una dimensión) y 121 (dos dimensiones); los datos de los arrays pueden ser de tipo cadena o numéricos siguiendo las mismas normas que para las variables correspondientes y los subíndices empiezan en 0, a menos que se especifique DIM 1, así pues, A(5) es el sexto elemento.

Los subíndices pueden ser variables enteras, constantes enteras o cualquier expresión numérica cuyo resultado sea, o pueda ser redondeado, un número entero. En las expresiones de arrays bidimensionales, el primer subíndice es la fila y el segundo la columna del array. Los arrays de cadenas sólo tienen una dimensión y por lo tanto sólo necesitan un subíndice.

Se pueden redimensionar los arrays durante la ejecución del programa pero, en algunos casos, esto conlleva bastantes problemas. Las dimensiones de un array pueden ser DINAMICAS (DYNAMIC), es decir, el valor de los subíndices para las variables se introducen en tiempo de ejecución.

Expresión numérica

Una expresión numérica es un grupo de números, variables numéricas, funciones y arrays ligados por concatenaciones aritméticas o

de tipo cadena o por operadores relacionales y lógicos; la secuencia de evaluación de una expresión numérica va de izquierda a derecha y en el siguiente orden (por regla general):

- expresiones entre paréntesis, empezando por el más interior;
- más y menos unarios;
- exponenciación (de izquierda a derecha si es múltiple);
- multiplicación y división;
- suma y resta.

Constante

Una constante puede ser un valor número o un literal tipo cadena.

Las constantes numéricas (NUMERIC) sirven para asignar valores a variables y arrays numéricos y siguen las mismas reglas.

Una constante CADENA (STRING) es cualquier secuencia de caracteres ASCII escrita entre comillas.

Nombre de fichero/número de fichero/ número de canal

A un conjunto de datos se le puede asignar un nombre de fichero (FILENAME) que consiste en una combinación de caracteres alfanuméricos, debiendo ser el primero de ellos una letra. El número de caracteres del nombre puede estar limitado (8 como mínimo, 238 como máximo). Por regla general, a cada nombre de fichero se le asigna un número de canal o referencia empezando por 1; cada número es único y si a un segundo fichero se le asigna un número que ya está siendo utilizado, entonces se cerrará el primer fichero y ya no estará disponible (normalmente).

En algunas versiones del BASIC podemos especificar el nombre de fichero/número/canal como una expresión numérica dentro de una sentencia, en otras sólo se permiten los enteros, en ese caso las expresiones numéricas se evaluarán y se redondeará el resultado si es preciso, antes de utilizarlas en la sentencia.

Fichero

Un fichero es un conjunto de datos utilizados como entrada o salida de un programa. Como ejemplo citaré los ficheros de bases de datos, los ficheros de índices, ficheros maestros y secundarios, ficheros de imagen de memoria (para los programas SAVED), ficheros de imagen de impresión, y ficheros tipo listado (para los programas LISTED y las salidas formateadas), ficheros de tablas (asociados al mantenimiento de ficheros), directorios de usuario y particiones del disco, ficheros temporales de trabajo, y los de dispositivo (por ejemplo, la impresora de líneas o el teclado) sólo por nombrar unos pocos.

Registro

Un fichero se compone de un número de registros; cada registro se compone de un grupo de uno o más items de datos.

Organización

La organización de un fichero determina el modo en el cual se van a poder leer o escribir los datos desde el dispositivo lógico que almacena dichos datos. Los métodos estándares de organización son:

SECUENCIAL (SEQUENTIAL) en la que los datos se escriben en el mismo orden en que se leen, es decir empezando por el principio y terminando por el final. En algunos sistemas los datos se leen del disco con las mismas combinaciones de longitud de registros con que fueron previamente grabadas; en otros las operaciones READ/WRITE (y sinónimos) son completamente independientes. La longitud de cada registro puede ser de un número fijo o variable de caracteres.

AL AZAR (RANDOM) en la que a cada registro se le asigna una única posición del disco, permitiendo una secuencia aleatoria para su lectura y escritura; la dirección, que es la dirección física del disco, se determina normalmente convirtiendo un descriptor alfanumérico del registro (un identificador de item, por ejemplo) en una dirección determinada del disco. Con esta organización la longitud fija de los registros es obligatoria y las operaciones READ/WRITE deben tener las mismas caracterís-

ticas de fichero. En algunos sistemas la longitud de registro tiene un tipo máximo (p. ej. 250 bytes); por lo menos en uno de los sistemas la longitud es fija y se limita a 16 caracteres.

INDEXADA (INDEXED) en la que se cataloga un descritor único de cada registro, en un fichero independiente de "claves", junto con la dirección física del disco que corresponde a cada registro; el fichero, en este caso, puede procesarse secuencialmente o al azar.

Antes y después del proceso se deben ABRIR (OPEN) y CERRAR (CLOSE) los ficheros; los parámetros de las sentencias OPEN y CLOSE determinan la forma en que se van a procesar los ficheros y aplican, si así se ha indicado, medidas de control y seguridad.

4. Sugerencias para la traducción de programas BASIC

a) Un aspecto importante de la conversión es si hay implicado un compilador de BASIC. Entonces cambian algunas características del dinamismo de variables y del programa fuente, tales como

CHAIN, DIM, COMMON y MERGE

b) En el ordenador IBM-PC se tiene además de lo indicado en el texto lo siguiente (por ejemplo):

BLOAD
Carga en memoria un programa binario desde memoria auxiliar (ver (BSAVE).

COM (n)
Activa o desactiva un adaptador de comunicaciones.

CSRLIN
Ver YPOS.

DEFSEG
Para conseguir direccionamientos en un segmento (64 K) dado.

EOF
Ver EOF$.

FILES
Da los nombres de ficheros de un disquete.

INPUT #
Lee datos de un fichero secuencial.

KEY (n)
Activa o desactiva la condición ON de una tecla dada.

LPOS
Devuelve posición dentro de la línea de impresión.

LPRINT USING
Ver PRINT USING.

OCT$
Da una tira con el valor octal de un número decimal o argumento.

PEN
Lee el lápiz luminoso.

SYSTEM
Acaba el BASIC y vuelve al DOS.

WIDTH
Define la longitud en número de caracteres de la línea de salida.

WRITE #
Escribe datos en un fichero secuencial.

4.1. Familiarícese con su ordenador, y con la documentación de los fabricantes sobre el BASIC y su ordenador.

4.2. Apunte en el Diccionario la página exacta del manual del fabricante en la que se encuentra la "descripción en profundidad de las sentencias" incluyendo aclaraciones abreviadas, si es preciso; con ello conseguirá que el proceso de traducción sea rápido sobre todo cuando el Indice del manual del fabricante no esté bien organizado o si sus descripciones son largas o poco claras.

4.3. Intente comprender el programa que va a traducir, no sólo la función principal sino también cómo se llega a ella y si es posible hágase con una información de salida ejecutando el programa con algunos datos de prueba.

4.4. Si lo considera necesario, organice el programa que va a traducir en un formato que sea cómodo de leer. Es preferible que cada línea contenga una sola sentencia. Divida el programa en bloques significativos e inserte comentarios que clarifiquen el significado de cada bloque (se necesitará memoria adicional, pero una buena documentación ayuda mucho). Renumere el programa permitiendo la inserción de líneas; señale específicamente todas aquellas instrucciones que hay que traducir antes de empezar la traducción.

4.5. Antes de empezar la traducción, revise algunas características del programa en su "totalidad" ya que puede existir cierta dependencia entre algunas secciones:

a) Ubicación de variables

- criterios de nomenclatura para especificar los tipos de datos;
- tamaños asumidos si no se utilizan sentencias DIM;
- número de caracteres de los nombres de datos.

b) Puntuación y caracteres especiales

- coma, punto y coma, punto decimal (coma decimal);
- $ y barra inclinada en las cadenas;
- valores de cadenas con/sin comillas;
- expresiones numéricas, enteros y posiciones de la pantalla con/sin paréntesis;
- juego de caracteres utilizado por el programa, por su impresora y por el ordenador (ASCII o SCREEN - DISPLAY);
- criterios para el número de fichero/canal;
- utilización del número 0 y la letra O;
- utilización de mayúsculas/minúsculas.

c) Reglas aritméticas

- secuencia de prioridades para las operaciones aritméticas;
- opción BASE para la numeración de los elementos de los arrays;
- valores de MODE asumidos para los tipos de datos, grados/radianes y juego de caracteres ASCII/EBCDIC.

d) Tratamiento de arrays

- tratamiento de subcadenas;
- requerimientos de memoria para grandes arrays.

e) Sentencias directamente relacionadas con la máquina

- valores asignados a listas de códigos para colores, siluetas y sonidos;
- referencia a la posición de pantalla, en modalidad gráficos (X,Y; X,Y relativas, posición P);
- instrucciones tipo PEEK y POKE que se refiren a una memoria física mayor que la de su ordenador o que ac-

ceden a elementos que no están disponibles a su sistema o que se encuentran ubicados en una posición distinta de memoria;

- utilización de expresiones numéricas en lugar de nombres/números de ficheros y enteros;
- utilización de ficheros de datos externos.

f) Abreviaturas y funciones con significado duplicado

- extraiga del Diccionario la expresión íntegra para las instrucciones abreviadas;
- revise las instrucciones más importantes del Diccionario para asegurarse de que se comportan como Vd. piensa.

4.6. Después de esta revisión, documente cuidadosamente los pasos que va a dar, bien en listados independientes o si hay suficiente memoria, con sentencias REM en el programa; guarde una copia de seguridad de su programa original.

4.7. Comience la traducción pasando un "test" a su programa. Si falla alguna instrucción mire la traducción correspondiente en el Diccionario, vuelva a codificarla (inserte un REM detallando las instrucciones originales, si la primera solución no le ha funcionado).

Continúe con esta operación hasta que no haya errores de ejecución.

4.8. Si en este momento el programa no funciona satisfactoriamente, verifique los datos de prueba y asegúrese de que son consistentes con lo que se espera que haga el programa ¡especialmente si es Vd. quien los introdujo desde el teclado!

4.9. Si los datos de prueba son válidos, intente ejecutar el programa a través de un listado impreso, apuntando los valores que van tomando las variables principales, incluso cuando éstos se asumen. Si el programa BASIC tiene una función TRACE, actívela para el programa completo o para bloques de sentencias/variables seleccionadas; en caso contrario inserte sentencias PRINT/DISP para sacar a impresión las variables principales y los resultados de las condiciones de comprobación críticas. A fin de minimizar las salidas a impresión producidas por TRACE es preferible que no utilice muchos datos.

4.10. Si el sistema se "destroza", es decir, si se pierde el cursor, verifique de nuevo las direcciones PEEK/POKE, es posible que se esté ejecutando un POKE sobre un área no permitida.

4.11. Si el programa sigue sin funcionar puede ser debido a un ERROR BASICO:

- Vd. ha cometido algún fallo en la revisión inicial (misma escritura con diferente significado, estándares asumidos distintos o errores similares);
- ha introducido algún error en la traducción en el sentido de que para una instrucción determinada el resultado es diferente.

Llegados a este punto Vd. necesita ayuda externa; resulta sorprendente ver cómo se puede solucionar un problema simplemente explicándoselo a alguien.

4.12. Aunque una codificación buena, clara y bien documentada ayuda muchísimo, es posible que Vd. no tenga suficiente memoria. Si tiene que hacer el programa más pequeño, tenga en cuenta las siguientes sugerencias:

- escriba varias sentencias en una línea ya que la numeración de líneas consume bytes;
- borre los espacios innecesarios;
- borre las sentencias REM; utilizan tantos bytes como caracteres tiene el comentario más uno (por lo menos);
- utilice variables en lugar de constantes; con ello ahorrará memoria y aumentará la velocidad de ejecución (a veces espectacularmente);
- reutilice las variables siempre y cuando la naturaleza de su utilización sea temporal (p. ej. la respuesta desde un terminal en diferentes puntos del programa);
- establezca subrutinas para grupos de sentencias del programa que realicen funciones idénticas; emplee la sentencia GOSUB para ejecutar la subrutina;
- utilice los elementos cero de las matrices;
- aplique niveles razonables para las dimensiones de las matrices (con 26 se cubre todo el alfabeto, pero es posible que a Vd. sólo le interesen las letras B, A, S, I y C.

4.13. Finalmente, ahora que ya ha conseguido traducir con éxito su programa, actualice el Diccionario con los pasos detallados de lo que ha tenido que hacer (¡le será útil para referencias posteriores!).

5. Visión general de Códigos para Micros

Dicho en pocas palabras, la sentencia PEEK (y sinónimos) examina el contenido de una posición determinada de memoria y devuelve la representación decimal ASCII del contenido de la posición, que es un número comprendido entre 0 y 255. En algunas máquinas un PEEK de áreas del MONITOR o del STACK (PILA) devuelve un valor estándar y nada significativo, ya que los diseñadores no quieren que se pueda consultar sus contenidos.

La sentencia POKE (y sinónimos) es la opuesta a PEEK, ya que coloca un valor (de 0 a 255) en la posición de memoria especificada. No se permite un POKE dirigido a la memoria ROM (Read Only Memory). Hay que poner mucho cuidado en los POKE dirigidos hacia otras áreas porque al modificar la memoria se puede provocar un mal funcionamiento del ordenador.

Se puede obtener mucho provecho de los PEEK y POKE dirigidos hacia el área de vídeo RAM (Random Access Memory) para aumentar la velocidad de las visualizaciones y para controlar los caracteres en pantalla; el procesador está continuamente monitorizando las áreas RAM haciendo aparecer en pantalla en forma inmediata los cambios efectuados en la zona de memoria reservada para las visualizaciones de formatos de pantalla. De forma similar, se puede "investigar", mediante un POKE, cualquier carácter de entrada en la pantalla para desencadenar una acción específica.

Las sentencias PEEK y POKE se utilizan también para producir reacciones físicas, como funciones de tabulación (TAB) en la pantalla, generación de sonidos, inhibición de condiciones de error, cambios de códigos de color y similares.

Se pueden construir rutinas de usuario que utilicen lenguaje máquina, efectuando POKEs sobre las instrucciones afectadas, directamente en memoria. En algunos Basics se puede utilizar la instrucción POKE para ejecutar la rutina de usuario.

En las siguientes Secciones se exponen ejemplos de direcciones de PEEK/POKE para unas máquinas determinadas. Las instrucciones PEEK/POKE vienen documentadas con mapas de memoria para dar una idea de las áreas de memoria que se pueden examinar/manipular

y con una descripción de la intencionalidad de dichas sentencias. Las memorias se han estandarizado a 64 Ks. La ubicación y el tamaño de las secciones de memoria pueden variar si el tamaño total de la memoria cambia; incluso los mapas de memoria de los distintos modelos de un mismo ordenador pueden variar considerablemente. Por consiguiente, los ejemplos que se muestran a continuación deben tomarse como una indicación de lo que puede suceder en un programa que Vd. está intentando traducir para que funcione en su ordenador.

También se incluyen ejemplos de Códigos para Gráficos en Color y Generadores de Sonido. Observará que los valores de los códigos de color varían de un ordenador a otro tanto como los colores que visualizan. De nuevo le recordamos que solamente son ejemplos y que son válidos únicamente para aportarle algunas ideas sobre posibles soluciones de los problemas detectados en la traducción.

Un elemento diferencial entre microordenadores es el juego de caracteres. Al final de esta Sección he incluido una tabla comparativa para evidenciar el amplio espectro y la gran variedad de caracteres disponibles. Los caracteres Ascii son casi estándar, aunque no del todo; fuera de este entorno no podemos hablar en absoluto de estandarización. Si el conjunto de caracteres representase una parte importante del programa que Vd. está traduciendo deberá conseguir una copia del juego de caracteres de la máquina para la que inicialmente se escribió el programa.

Hay varias razones para esto:

1. No es extraño que varíe el juego de caracteres para una máquina determinada dependiendo de la fecha de fabricación, área de ventas o sistema operativo.
2. De hecho, la mayoría de las máquinas disponen de dos juegos de caracteres: el DISPLAY (utilizado en la memoria de la pantalla) y el CHARACTER (utilizado en la memoria principal RAM).
3. Las tablas que he representado corresponden a sistemas que han podido quedarse desactualizados o que no estén disponibles cuando Vd. los necesite.

Las tablas representan el juego CHARACTER. Los juegos DISPLAY son similares a los correspondientes juegos CHARACTER en el sentido de que contienen caracteres similares pero con diferentes valores decimales. Por ejemplo:

A = 01000001 = 65 en memoria principal

A = 00000001 = 1 en memoria de pantalla

El rango de lo que se puede "traducir" tanto del juego de caracteres de Display como del de Character puede variar de una máquina a otra. No resulta raro que algunas máquinas tengan un gran número de caracteres único para un juego.

Por último añadiré que en esta Sección sólo se ha considerado un número pequeño de ordenadores (seleccionados al azar); se podría haber realizado el mismo ejercicio con otros muchos, pero habría supuesto un esfuerzo considerable y no habría proporcionado grandes beneficios. Sin embargo, se ha tenido en consideración una amplísima gama de versiones del Basic para la preparación de la Sección Alfabética que define las distintas sentencias, comandos y funciones.

5.1. EJEMPLOS SOBRE PEEK Y POKE

5.1.1. Ejemplos de PEEK y POKE en el APPLE II

Dirección Decimal	Valor retornado por PEEK/Acción tomada por POKE
32	margen izquierdo de la ventana de texto (0-39)
33	anchura de la ventana de texto (1-40)
34	margen superior de la ventana de texto
35	margen inferior de la ventana de texto
36	posición horizontal del cursor
37	posición vertical del cursor
50/127	activa modalidad de vídeo invertido (Integer BASIC)
50,255	activa modalidad de vídeo normal (Integer BASIC)
202/3	dirección de HIMEM (pos. alta de memoria)
204/5	dirección de LOMEM (pos. baja de memoria)
216	ha ocurrido una condición ON ERR GO TO si 128-255
216,X	ha ocurrido una condición ON ERR GO TO desactivado si 1-127
218/9	especifica la línea de programa con la condición de error
222	especifica el tipo de error

−16384/49152	lee el teclado; la tecla pulsada si el valor es 128-288; la tecla actual (ascii) si el valor del PEEK es menor que 128
−16368/49168	restaura la señal del teclado a cero para que se pueda leer el siguiente carácter
−16352/49200	provoca un "click" en el altavoz interno
−16336/49284	provoca un "click" en el conector del casete
−16304/49231/2	selecciona modalidad GRAFICOS o TEXTO
−16302/49234	selecciona gráficos en pantalla completa
−16301/49235	selecciona gráficos y texto
−16300/49236	selecciona PAGINA 1 (gráficos o texto)
−16299/49237	selecciona PAGINA 2 (gráficos o texto)
−16298/49238	selecciona gráficos en baja resolución
−16297/49239	selecciona gráficos en alta resolución
−16296/49240/1	conecta/desconecta el control 0 de juegos
−16294/49242/3	conecta/desconecta el control 1 de juegos
−16292/49244/5	conecta/desconecta el control 2 de juegos
−16290/49246/7	conecta/desconecta el control 3 de juegos
−16287/49249	control de juego 0, se ha pulsado el botón si 128-255
−16286/49250	control de juego 1, se ha pulsado el botón si 128-255
−16285/49251	control de juego 2, se ha pulsado el botón si 128-255
−16272/49264	desconecta el voltaje de la señal de salida durante 0,5 microsegundos
− 151	punto de entrada para el MONITOR
− 182	salva el contenido del registro
− 193	restaura el contenido del registro
− 198	sonido (beep) en el altavoz incorporado
− 211	imprime el mensaje ERR y produce un beep en el altavoz incorporado
− 336	retorna al BASIC
− 380	activa modalidad de vídeo normal
− 384	activa modalidad de vídeo invertido
− 550	imprime el valor contenido en el registro Acumulador
− 715	en espera de la pulsación de una tecla
− 756	en espera de la pulsación de una tecla (ignora caracteres de control (escape))

– 856	ejecuta un bucle de retardo
– 912	desplaza una línea hacia arriba la ventana de texto
– 922	mueve el cursor una línea hacia abajo
– 926	retorno de carro y salto de línea en pantalla
– 936	limpia la pantalla y sitúa el cursor en la esquina superior izquierda
– 954	limpia la pantalla desde la posición especificada hasta el final hacia abajo
– 958	limpia la pantalla desde la posición actual hasta el final hacia abajo
– 998	mueve el cursor una línea hacia arriba
– 1008	manda a la pantalla un carácter de retroceso
– 1052	beep en el altavoz del APPLE II durante 1/10 seg.
– 1063	envía el carácter de "campana"
– 1718	envía 1/256 espacios, según el contenido del registro A, al dispositivo de salida
– 1720	envía 3 espacios al dispositivo de salida
– 1724	imprime el contenido del registro X
– 1727	imprime el contenido de los registros X y A
– 1728	imprime el contenido de los registros Y y X
– 1985	incrementa en 3 el color actual en gráficos baja resolución
– 1998	pone todas las filas (baja resolución) en negro o rellena la pantalla con "@"

Nota: Las direcciones mayores que 32767 se expresan como números negativos iguales a la dirección menos 65536. Algunas de las funciones descritas anteriormente se activan por el simple hecho de acceder a la dirección; por ejemplo PEEK-16352 provoca un "click" en el altavoz interno; un POKE provoca dos "clicks".

5.1.2. Ejemplos de PEEK y POKE en el PET

Dirección Decimal	Valor retornado por PEEK/Acción tomada por POKE
122/3	comienzo del texto BASIC (puntero de dirección)
124/5	comienzo de variables
126/7	final de variables
128/9	final de tabla de arrays

130/1	comienzo de cadenas
132/3	inicio de memoria
134/5	dirección más alta de RAM
138/9	número de línea para CONTINUE
140/1	próxima sentencia a ejecutar
142/3	números de líneas de datos para errores
144/5	sentencia de datos
243,122	antes de OPEN para inicializar el buffer TAPE-1
244,2	antes de OPEN para inicializar el buffer TAPE-1
243,58	antes de OPEN para inicializar el buffer TAPE-2
244,3	antes de OPEN para inicializar el buffer TAPE-2
59459,255	selecciona modo salida en todas las líneas
59459,0	selecciona modo entrada en todas las líneas.
	Se pueden conseguir distintas combinaciones de entrada/salida en las líneas dependiendo del valor ASCII correspondiente a los bits que se hayan activado (00000000 - 11111111)

5.1.3. Ejemplos de PEEK y POKE en el SHARP MZ80K

Dirección Decimal	Valor retornado por PEED/Acción tomada por POKE
4360,0	desactiva SAVE, LIST
10680,0	desactiva SAVE, LIST
10681,0	desactiva SAVE, LIST en la recarga
10682,1	antes del SAVE, el programa se ejecuta automáticamente al volverlo a cargar
4464,0/1	cambio mayúsculas/minúsculas
4465,X	mueve el cursor X posiciones a la derecha (1-40)
4466,Y	mueve el cursor Y líneas hacia abajo (1-24)
6636,0	desactiva la tecla BREAK (como seguridad)
8767,0	desactiva la tecla BREAK (como seguridad)
8768,0	desactiva la tecla BREAK (como seguridad)
8769,0	desactiva la tecla BREAK (como seguridad)
6636,205	activa la tecla BREAK
8767,218	activa la tecla BREAK

8768,133	activa la tecla BREAK
8769,19	activa la tecla BREAK
10167,1	permite acceder a la zona no vídeo RAM
17828	permite la entrada continuada desde el teclado
18440,0 18441,0	cambia la línea 1 a 0 para que no pueda ser modificada o borrada
59555,0/1	apaga/enciende el CRT (la pantalla)
18859 18860	utilizado para seleccionar unidades de disco para entrada/salida de ficheros de datos en disco

5.1.4. Ejemplos de PEEK y POKE en el SINCLAIR ZX81

Dirección Decimal	Valor retornado por PEEK/Acción tomada por POKE
16388/9	dirección del primer byte después del último byte existente en la memoria RAM (RAMTOP)
16396/7	dirección de comienzo del fichero de visualización (D - FILE)
16398/9	dirección de la posición de PRINT en el fichero de visualización (DF - CC)
16400/1	dirección de comienzo del fichero de variables (VARS)
16434/5	origen para el generador de números al azar (SEED)
16436/7	contador de páginas visualizadas en la pantalla de TV. (FRAMES)

5.1.5. Ejemplos de PEEK y POKE en el TANDY TRS80 MODEL III

Dirección Decimal	Valor retornado por PEEK/Acción tomada por POKE
16412,1	cursor parpadeante o fijo
16419,176	código para el carácter de cursor
16916,X	desplaza X líneas de la pantalla
16888	código para el valor de banda E/S serie
16890	conmutador Espera/No espera E/S serie
16872	buffer de caracteres de entrada

16880	buffer de caracteres de salida
16889	paridad/longitud de palabra/bit de stop
16928/16931	ORIGEN y DESTINO
16919/16924	reloj de tiempo real

5.2. MAPAS DE MEMORIA Y CONVERSION HEXADECIMAL/DECIMAL

5.2.1. Mapa de Memoria simplificado del APPLE II

Dirección hexadecimal		Dirección decimal
0		0
100	PROGRAMAS DEL SISTEMA	256
200	PILA DEL SISTEMA	512
300	BUFFER DE ENTRADA DEL TECLADO	768
400	POSICIONES DEL VECTOR MONITOR	1024
800	TEXTO Y GRAFICOS EN BAJA RESOLUCION PAGINA 1	2048
C00	TEXTO Y GRAFICOS EN BAJA RESOLUCION PAGINA 2	3072
2000	LIBRE	8192
4000	GRAFICOS EN ALTA RESOLUCION - PAGINA 1	16384
6000	GRAFICOS EN ALTA RESOLUCION - PAGINA 2	24576
C000	LIBRE	49152
D000	ENTRADA/SALIDA	53248
FFFF	ROM PARA BASIC	65535

5.2.2. Mapa de Memoria simplificado del BBC

Dirección hexadecimal		Dirección decimal
0		0
100	PAGINA CERO	256
200	PILA 6602	512
300	AREA DE TRABAJO DEL SISTEMA OPERATIVO	768
400	AREA DE TRABAJO SISTEMA OPERATIVO Y VIDEO	1024
800	AREA DE TRABAJO DEL LENGUAJE	2048
900	BUFFER DE SONIDO Y ENVOLVENTE	2304
A00	BUFFER DE TRANSMISION RS423	2560
B00	BUFFER DE RECEPCION RS423	2816
C00	BUFFER DE CLAVES SOFT	3072
D00	DEFINICION 1 DE CARACTERES	3328
F00	AREA DE TRABAJO DEL DISCO Y RED	3840
1000	AREA DE TRABAJO PARA ROM	4096
1600	DEFINICION 2 DE CARACTERES	5632
	AREA DE TRABAJO DEL DISCO	
2000	BASIC Y PROGRAMA DEL USUARIO	8192
	ALMACENAMIENTO DINAMICO DE VARIABLES	
4000/ 8000	RAM ALTA RESOLUCION	16384/ 32768
C000	AREAS ROM PAGINADAS	49152
F000	SISTEMA OPERATIVO ROM	64512
FF00	MEMORIA MAPEADA PARA ENTRADA/SALIDA	65280
FFFF	SISTEMA OPERATIVO ROM	65535

5.2.3. Mapa de Memoria simplificado del PET

Dirección hexadecimal		Dirección decimal
0		0
0100	ALMACENAMIENTO DE TRABAJO BASIC SISTEMA OPERATIVO (SO)	256
0200	PILA	512
0300	ALMACENAMIENTO DE TRABAJO DEL SO	768
0400	BUFFERES DEL CASETE	1024
1000	AREA DE TEXTO BASIC	4096
2000	ALMACENAMIENTO DE VARIABLES DE PRUEBAS	8192
8000	EXPANSION RAM (6 X 4 K)	32768
9000	PANTALLA RAM	36864
C000	EXPANSION ROM (3 X 4 K)	49152
D000	ROM-1 BASIC	53248
E000	ROM-2 BASIC	57344
E800	EDITOR DE PANTALLA	59392
E810	ENTRADA/SALIDA DEL PET	59408
E820	TECLADO	59424
E840	IEEE - 488 BUS	59456
F000	CASETE DE USUARIO – PUERTA	61440
FFFF	DIAGNOSTICOS DEL SO	65535

5.2.4. Mapa de Memoria simplificado del SHARP MZ-80K

Dirección hexadecimal		Dirección decimal
0		0
1000	MONITOR	4096
1200	PILA Y AREAS DE TRABAJO	4608
6000	BASIC Y PROGRAMA DE USUARIO	24576
D000	EXPANSION DE MEMORIA	53248
E000	RAM VIDEO	57344
FFFF	CONTROL DE PERIFERICOS	65535

5.2.5. Mapa de Memoria simplificado del TANDY TRS80 MODEL I

Dirección hexadecimal		Dirección decimal
0		0
3000	ROM NIVEL II	12288
3C00	RESERVADO PARA MEMORIA MAPEADA DE E/S	15360
4000	MEMORIA DE VIDEO	16384
4015	VECTORES BASIC (RST's 1-7)	16405
401D	TECLADO DCB	16413
4025	VIDEO DCB	16421
402D	IMPRESORA DE LINEAS	16429
41E6	RESERVADO	16870
42E9	BUFFER E/S	17129
	TEXTO PROGRAMA BASIC	
	ARRAYS	
	MEMORIA LIBRE	
	PILA	
	AREA PARA CADENAS	
	RESERVADO CON OPCION DE TAMAÑO DE MEMORIA	
4FFF	FIN SISTEMA DE "4K"	20479
7FFF	"16K"	32767
BFFF	"32K"	49151
FFFF	"48K"	65535

5.2.6. Mapa de Memoria simplificado TANTY TRS80 MODEL II

Dirección hexadecimal		Dirección decimal
0		0
2800	AREA RESERVADA DEL SISTEMA	10240
3000	AREA DE "OVERLAYS ALTOS" DEL TRSDOS COMPARTIDA CON POSIBLES USUARIOS	12288
TOP	AREA DE USUARIOS	TOP
7FFF	PUEDE ESTAR RESERVADA POR EL TRSDOS PARA PROGRAMACION ESPECIAL	32767
FFF	EXPANSION	65535

5.2.7. Mapa de Memoria simplificado del TANDY TRS80 MODEL III

Dirección hexadecimal		Dirección decimal
0		0
4000	ROM	16384
5200	TRSDOS	20992
5500	DEBUG	21760
7000	OVERLAYS DEL DOS	28672
8000	FORMAT	32768
	BASIC	
	PROGRAMAS DE USUARIO	
FFFF	EXPANSION	65535

5.2.8. Mapa de Memoria simplificado del TRS80 COLOR AMPLIADO

Dirección hexadecimal		Dirección decimal
0		0
100	PAGINA RAM DIRECTA S.O.	256
400	PAGINA RAM AMPLIADA S.O.	1024
600	MEMORIA DE PANTALLA EN MODALIDAD TEXTO	1536
3600	MEMORIA PANTALLA EN GRAFICOS (8 X 1,5 K)	13824
4000	PROGRAMA DE USUARIO Y VARIABLES	16384
8000	AREA DE EXPANSION	37768
A000	BASIC - COLOR AMPLIADO	40960
C000	BASIC - COLOR	49152
FF00	MEMORIA CARTUCHO	65280
FFFF	ENTRADA SALIDA	65535

5.2.9. Mapa de Memoria simplicado del SINCLAIR ZX81

Dirección hexadecimal		Dirección decimal
0		0
2000	8K-ROM PARA INTERPRETE BASIC Y MONITOR	8192
4000	SIN UTILIZAR	16384
	MEMORIA RAM DE LOS USUARIOS	
	VARIABLES DEL SISTEMA ZX81	
	PROGRAMA	D-FILE
	FICHERO DE VISUALIZACION	VARS
	VARIABLES	E-LINE
	LINEA QUE SE ESTA TECLEANDO Y AREA DE TRABAJO	STK BOT
	PILA PARA CALCULOS	STK END
	LIBRE	
	PILA DE LA MAQUINA	ERR-SP
	PILA PARA GOSUB	RAM TOP
	RUTINAS USUARIO	
7FFF	AREA DE EXPANSION	32767

5.2.10. Mapa de Memoria simplificado del VIC

Dirección hexadecimal		Dirección decimal
0		0
3FF	1K RAM	1023
FFF	3K EXPANSION RAM	4095
1DFF	4K RAM	7679
1FFF	PANTALLA RAM	8191
3FFF	8K EXPANSION RAM/ROM	16383
5FFF	8K EXPANSION RAM/ROM	24575
7FFF	8K EXPANSION RAM/ROM	32767
8FFF	4K ROM GENERADOR DE CARACTERES	38863
90FF	6561	37119
95FF	6522 No. 1 & No. 2	38399
97FF	COLOR RAM	38911
9BFF	EXPANSION E/S No. 1	39935
9FFF	EXPANSION E/S No. 2	40959
BFFF	8K EXPANSION ROM/RAM	49151
DFFF	8K ROM BASIC	57343
FFFF	8K ROM KERNEL	65535

CONVERSION HEXADECIMAL/DECIMAL

HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0		0		0		0		0	
1	1.048.576	1	65.536	1	4.096	1	256	1	16	1	1
2	2.097.152	2	131.072	2	8.192	2	512	2	32	2	2
3	3.145.728	3	196.608	3	12.288	3	768	3	48	3	3
4	4.194.304	4	262.144	4	16.384	4	1.024	4	64	4	4
5	5.242.880	5	327.680	5	20.480	5	1.280	5	80	5	5
6	6.291.456	6	393.216	6	24.576	6	1.536	6	96	6	6
7	7.340.032	7	458.752	7	28.672	7	1.792	7	112	7	7
8	8.388.608	8	524.288	8	32.768	8	2.048	8	128	8	8
9	9.437.184	9	589.824	9	36.864	9	2.304	9	144	9	9
A	10.485.760	A	655.360	A	40.960	A	2.560	A	160	A	10
B	11.534.336	B	720.896	B	45.056	B	2.816	B	176	B	11
C	12.582.912	C	786.432	C	49.152	C	3.072	C	192	C	12
D	13.631.488	D	851.968	D	53.248	D	3.328	D	208	D	13
E	14.680.064	E	917.504	E	57.344	E	3.584	E	224	E	14
F	15.728.640	F	983.040	F	61.440	F	3.840	F	240	F	15

5.3. CODIGOS PARA EL COLOR

5.3.1. Ejemplos de códigos de color en gráficos del APPLE II

Código color	Baja resolución	Alta resolución
0	Negro	Negro
1	Magenta	Verde
2	Azul oscuro	Violeta
3	Morado	Blanco
4	Verde oscuro	Negro
5	Gris - 1	Naranja
6	Azul medio	Azul
7	Azul claro	Blanco
8	Marrón	
9	Naranja	
10	Gris - 2	
11	Rosa	
12	Verde claro	
13	Amarillo	
14	Turquesa claro	
15	Blanco	

5.3.2. Ejemplos de códigos de color en gráficos del ACORN ATOM

Código color		Combinaciones color/modalidad		
Fondo	Primer Plano	2	1;5	0, 3, 4, 6
128	0	Negro	Negro	Negro
129	1	Rojo	Rojo	Blanco
130	2	Verde	Amarillo	
131	3	Amarillo	Blanco	
132	4	Azul		
133	5	Magenta		
134	6	Gris azulado (Cian)		
135	7	Blanco		
136 - 143	8 - 15	Destellos (0-7) Destellos (128-135)		

Parámetros para control del color en pantalla del ACORN ATOM

Modo	Memoria	Colores	Puntos de gráficos	Caracteres de texto
0	20K	2	640 X 256	80 X 32
1	20K	4	320 X 256	40 X 32
2	20K	16	160 X 256	20 X 32
3	16K	2	–	80 X 25
4	10K	2	320 X 256	40 X 32
5	10K	4	160 X 256	20 X 32
6	8K	2	–	40 X 25
7	1K	–	–	40 X 25

5.3.3. Ejemplos de códigos de color en graficos del PET

Código color		
Fondo	Primer plano	Color
0	0	Negro
16	1	Rojo (Oscuro)
32	2	Verde (Oscuro)
48	3	Amarillo (Oscuro)
64	4	Azul (Oscuro)
80	5	Magenta (Oscuro)
96	6	Gris azulado-Cian (Oscuro)
112	7	Blanco
128-240	8-15	Mismos colores que los de arriba pero con diferentes tonalidades (matices)

POKE	59471, X	fija los colores del primer plano y fondo; X es la suma de los códigos de color con valor comprendido entre 0-255
	59459,255	permite cambiar los colores de fondo y de primer plano
	59459,0	no permite cambiar los colores de primer plano/fondo
	59459,240	permite el cambio de los colores de fondo
	59459,15	permite el cambio de los colores de primer plano

5.3.4. Ejemplos de códigos de color en gráficos del VIC

Códigos de color

Borde	Toladidad 1 de pantalla		Tonalidad 2 de pantalla
0	8	Negro	–
1	24	Blanco	–
2	40	Rojo	–
3	56	Cian	184
4	72	Morado	200
5	88	Verde	216
6	104	Azul	232
7	120	Amarillo	248
–	136	Naranja	152
–	168	Rosa	–

POKE 36879,X fija los colores de la pantalla y el borde; X es la suma de los códigos de colores; las tonalidades 1 y 2 de pantalla son alternativas

5.4. CODIGOS PARA EL SONIDO

5.4.1. Ejemplos de generación de sonido en el VIC

Clave Musical	Valor de POKE	Valor de POKE (siguiente octava)	Valor de POKE (siguiente octava)
C	131	193	224
C sostenido	138	196	
D	145	200	
D sostenido	151	203	
E	157	206	
F	162	209	
F sostenido	167	211	
G	172	214	
G sostenido	177	216	
A	181	218	
A sostenido	185	220	
B	189	222	

36878,X	Controla el nivel de sonido para X comprendido entre 0 y 15
36874,Y	genera el TONO-A para Y comprendido entre 128 y 255
36875,Y	genera el TONO-B para Y comprendido entre 128 y 255
36876,Y	genera el TONO-C para Y comprendido entre 128 y 255
36877,Y	genera ruido Blanco para Y comprendido entre 128 y 255
	generador apagado con Y = 0

5.5. COMPARACION ENTRE LOS JUEGOS DE CARACTERES DE LOS MICROS

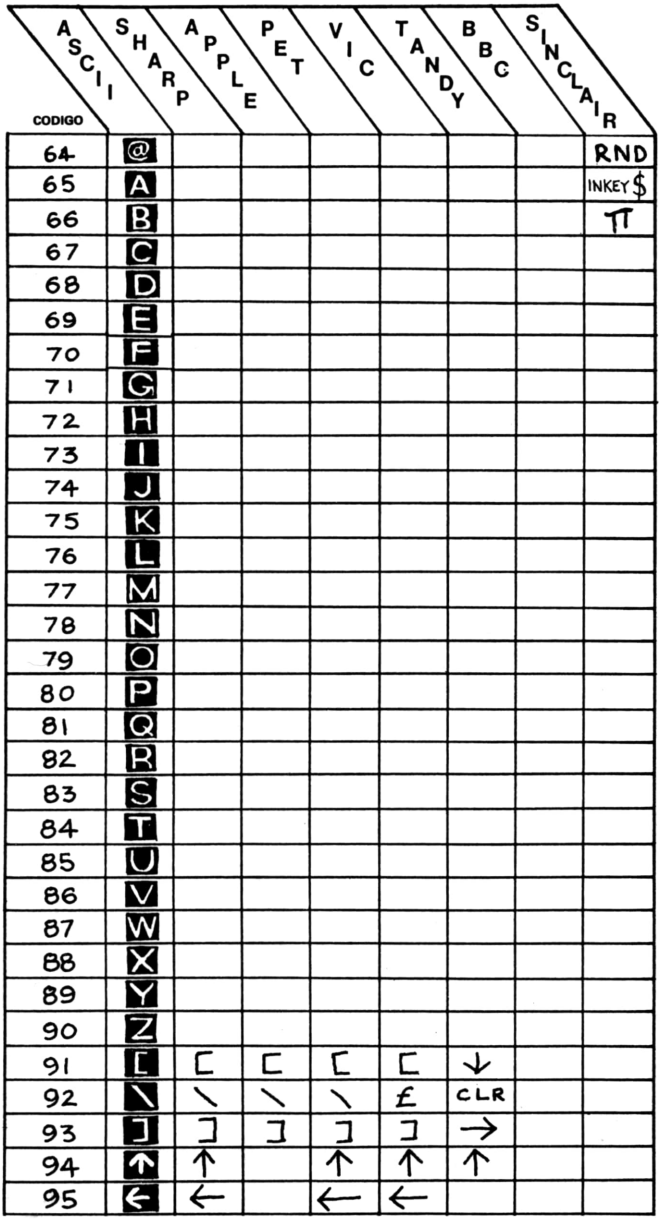
En las páginas siguientes se relacionan los distintos juegos de caracteres ASCII. Son caracteres que se pueden imprimir directamente o mediante la función CHR$. Algunas máquinas (por ejemplo, el Sharp) disponen de un juego adicional de caracteres (de visualización) para los comandos PEEK y POKE. En las tablas, asuma, si no se indica lo contrario, que el carácter de cualquier lista coincide con el carácter ASCII.

Nota: El PET tiene el juego ASCII de mayúsculas con gráficos, "o" minúsculas con mayúsculas. Sólo se ha listado aquí el juego de minúsculas con mayúsculas.

CARACTERES ASCII Y FUNCIONES DE CONTROL
(asumir ASCII si no se especifica nada en contra)

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR	
0					NUL			NUL	SP
1					SOH				
2					STX			ACTIVA IMPRES	
3					ETX		BRK	DESACT. IMPRES	
4					EOT				
5					ENQ				
6					ACK				
7				CAMPANA	BEL			BEEP	
8				CUR ←	BS		CUR ←	BS	
9					HT		CUR →	FS	
10				SALTO DE LINEA	LF		CUR ↓	LF	
11					VT			LF ↑	"
12					FF		LIMPIA	LIMPIA TEXTO	£
13				RETORNO DE CARRO	CR	CR	ENT	CR	$
14					SO	SWITCH LOWER		MODO PAG. ON	:
15					SI			MODO PAG. OFF	?
16					DLE			CLR GRAPH	(
17		CUR ↓			DC1	CUR ↓			)
18		CUR ↑			DC2	RVS ON			>
19		CUR →			DC3	CLR HOME			<
20		CUR ←			DC4	INST DEL			=
21			CUR →		NAK		CUR ←		+
22		CLR			SYN				-
23					ETB				*
24				CANCELA LINEA	CAN				/
25					EM				;
26					SUB				,
27					ESC				.
28					FS	ROJO			0
29					GS	CUR →			1
30					RS	VERDE		HOME	2
31					US	AZUL			3

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR
32	SP							4
33	!							5
34	"							6
35	#							7
36	$							8
37	%							9
38	&							A
39	'							B
40	(							C
41	)							D
42	*							E
43	+							F
44	,							G
45	-							H
46	.							I
47	/							J
48	0							K
49	1							L
50	2							M
51	3							N
52	4							O
53	5							P
54	6							Q
55	7							R
56	8							S
57	9							T
58	:							U
59	;							V
60	<							W
61	=							X
62	>							Y
63	?							Z


CODIGO
ASCII
SHARP
APPLE
PET
VIC
TANDY
BBC
SINCLAIR
64 @ RND
65 A INKEY $
66 B π
67 C
68 D
69 E
70 F
71 G
72 H
73 I
74 J
75 K
76 L
77 M
78 N
79 O
80 P
81 Q
82 R
83 S
84 T
85 U
86 V
87 W
88 X
89 Y
90 Z
91 [
92 \ £ CLR
93]
94 ↑
95 ←

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR	
96		SP						£	
97		a	a/A						
98		b	b/B						
99		c	c/C						
100		d	d/D						
101		e	e/E						
102		f	f/F						
103		g	g/G						
104		h	h/H						
105		i	i/I						
106		j	j/J						
107		k	k/K						
108		l	l/L						
109		m	m/M						
110		n	n/N						
111		o	o/O						
112		p	p/P						CUR ↑
113		q	q/Q						CUR ↓
114		r	r/R						CUR ←
115		s	s/S						CUR →
116		t	t/T						GRAFICO
117		u	u/U						EDITAR
118		v	v/V						NUEVA LINEA
119		w	w/W						CANCELAR
120		x	x/X						MODO
121		y	y/Y						FUNCION
122		z	z/Z						
123	o	{	[					¼	
124		|	|		|			ll	
125		}	]					¾	
126		~				π		÷	CONS-TANTE
127								?	CURSOR

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR
128			SP					
129								
130								
131								
132								
133						f1		
134						f3		
135				CAMPANA		f5		
136				CUR ←		f7		
137						f2		
138				LF		f4		
139						f6		
140						f8		
141				CR		SHIFT RET		
142						SWITCH LC		
143								
144						BLK		
145						CUR ↑		
146		e				RVS OFF		
147						CLR HOME		
148						INST DEL		
149				CUR →				
150		t						
151		g						
152		h		CANCELA LINEA				
153								
154		b						
155		x						
156		d				PURPURA		
157		r				CUR ←		
158		p				AMARI-LLO		
159		c				CIAN		

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR
160		q	SP		SP		SP	INV 4
161		a	!				!	INV 5
162		z	"				"	INV 6
163		w	#				#	INV 7
164		s	$				$	INV 8
165		u	%				%	INV 9
166		i	&				&	INV A
167			'				'	INV B
168		Ö	(				(	INV C
169		k	)				)	INV D
170		f	*				*	INV E
171		v	+				+	INV F
172			,				,	INV G
173		ü	-				-	INV H
174		ß	.				.	INV I
175		j	/				/	INV J
176		n	0				0	INV K
177			1				1	INV L
178		Ü	2				2	INV M
179		m	3				3	INV N
180			4				4	INV O
181			5				5	INV P
182			6				6	INV Q
183		o	7				7	INV R
184		l	8				8	INV S
185		Ä	9				9	INV T
186		ö	:				:	INV U
187		ä	;				;	INV V
188			<				<	INV W
189		y	=				=	INV X
190			>				>	INV Y
191			?				?	INV Z

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR	
192			SP	@				@	" "
193				A				A	
194				B				B	
195				C				C	
196				D				D	
197				E				E	
198				F				F	
199				G				G	
200				H				H	
201				I				I	
202				J				J	
203				K				K	
204				L				L	
205				M				M	
206				N				N	
207				O				O	
208				P				P	
209				Q				Q	
210				R				R	
211				S				S	
212				T				T	
213				U				U	
214				V				V	
215				W				W	
216				X				X	
217				Y				Y	
218				Z				Z	
219				[				[	
220				\				\	
221				]				]	
222								^	
223								_	

CODIGO	ASCII	SHARP	APPLE	PET	VIC	TANDY	BBC	SINCLAIR	
224				SP				£	
225				a/A				a	
226				b/B				b	
227				c/C				c	
228				d/D				d	
229				e/E				e	
230				f/F				f	
231				g/G				g	
232				h/H				h	
233				i/I				i	
234				j/J				j	
235				K/K				k	
236				l/L				l	
237				m/M				m	
238				n/N				n	
239				o/O				o	
240				P/P				p	
241				q/Q				q	
242				r/R				r	
243				s/S				s	
244				t/T				t	
245				u/U				u	
246				v/V				v	
247				w/W				w	
248				x/X				x	
249				y/Y				y	
250				z/z				z	
251				[				¼	
252				\				½	
253				]				¾	
254								÷	
255				π				INVERSE VIDEO	

6. Comentarios de los lectores

Al escribir este Diccionario he realizado un esfuerzo especial para asegurar la exactitud de las definiciones intentando que fuesen todo lo amplias que permiten las representaciones BASIC. Sin embargo, estoy seguro de que puede haber omisiones importantes y es posible que algunas de las definiciones necesiten una explicación más detallada o que estén mal enfocadas y se hubiesen podido simplificar. Agradeceré cualquier comentario que los usuarios (así como fabricantes y suministradores de software) quieran enviarme sobre estos puntos.

Independientemente de las críticas sobre el Diccionario, agradeceré los comentarios sobre el BASIC en general, relacionados con:

Inconsistencias	sentencias y/o parámetros que no tienen un significado estándar y no están representados en el Diccionario.
Fallos	sentencias BASIC que no funcionan en su ordenador.
Ausencias	características que están normalmente disponibles en la mayoría de los BASICs, pero no lo está en su versión.
Absurdos	soluciones recomendadas que son "absurdas" y que normalmente tratan de obviar una deficiencia del hardware o software estándar.
Peek/Poke	ejemplos de Peek, Poke, Mapas de Memoria, Códigos para el color y amplitud de valores para el sonido.
No estandarización	problemas asociados con el crecimiento del hardware y/o software.
Mala documentación	documentación sobre hardware o software, pobre o mal organizada.

Por último, si Vd. simplemente no consigue traducir un programa, hágamelo saber; algún lector puede haberme enviado ya una versión que funciona y me encantará cambiarla con Vd.

Envíe sus críticas, comentarios, problemas y soluciones a:

W. A. HART
BASIC SWAP
204 West George Street
GLASGOW

7. Relación alfabética de comandos, sentencias, funciones y operaciones del BASIC

A.–Ver ABS o AND.

ABORT.–Termina el programa actual inmediatamente por regla general sin cerrar ficheros o sin completar las sentencias de final de trabajo.

ABS (expresión numérica).–Devuelve el valor absoluto de la expresión numérica.

AC.–Ver ACS.

ACCESS.–Define la modalidad de ACCESO de un fichero de datos durante el OPEN; como READ solamente, WRITE solamente, MODIFY/UPDATE o APPEND (LEER, ESCRIBIR, MODIFICAR/ACTUALIZAR o AÑADIR al final).

ACL nombre de fichero, "ID de usuario", "atributos".–Especifica y relaciona los tipos de control de acceso al fichero (leer, escribir, ejecutar) para el fichero y el usuario especificado.

ACS (expresión numérica).–Retorna el valor principal del arco coseno de la expresión numérica en unidades angulares (radianes, no grados).

AD.–Ver ADVAL.

ADR (variable - A).–Devuelve la dirección de memoria de la variable especificada.

ADVAL (canal de entrada analógica).–Devuelve el valor digital (0-65520 por 16) de la tensión de entrada (0-2.5 voltios) para el canal especificado.

NOTAS DE TRADUCCION

ADVANCE *fichero; entero, variable numérica.—Salta un número de registros en un fichero formateado BASIC, hacia delante o hacia atrás dependiendo del signo del entero, la variable retorna el resultado de la operación de salto, si el final o comienzo del fichero se detecta antes de que se complete la operativa.

ALL.—Ver TRACE ALL.

ALLOW.—Especifica el tipo de acceso al fichero, permitido a otros programas en forma simultánea; las opciones son NONE (NINGUNO), READ (LEER) solamente y WRITE (ESCRIBIR) que incluye LEER.

APPEND nombre de programa.—AÑADE al programa actual en memoria, un programa fuente previamente almacenado en casete, cinta o disco.

ARRAYS.—Ver CLEAR CHARACTER ARRAYS.

AS.—Ver NAME.

ASC.—Ver ASCII.

ASCII (cadena).—Devuelve la representación decimal ASCII del primer carácter de la cadena especificada (0-255).

ASG.—Ver ASSIGN.

ASN (expresión numérica).—Devuelve el valor principal del arco seno de la expresión numérica en unidades angulares (radianes, no grados).

NOTAS DE TRADUCCION

ASSIGN especif. de fichero TO número de fichero, variable de retorno.—Abre un fichero de datos asignándole un número e indica el resultado de la apertura en la variable de retorno.

ASSIGN "fichero".—Reserva un dispositivo periférico para dedicárselo al programa que lanza el comando.

ATN (expresión numérica).—Devuelve el valor principal del arco tangente de la expresión numérica en unidades angulares (radianes, no grados).

AU.—Ver AUTO.

AUDIT nombre de fichero.—Copia todas las entradas y salidas sobre la consola, en el fichero especificado para AUDIT.

AUTO número de línea, valor del incremento.—Permite la numeración automática de todas las líneas del programa o empezando en el número de línea especificado de comienzo del programa.

AVAIL.—Devuelve el número total de registros disponibles en el dispositivo de almacenamiento masivo actual.

NOTAS DE TRADUCCION

B.–Ver BGET$.

BAPPEND.–AÑADE el programa binario especificado al programa que hay en memoria.

BASE 0 ó 1.–Establece el comienzo de la numeración de los elementos de un array en 0 ó 1 según se precise; normalmente la numeración empieza por 0, no por 1.

BEEP.–Genera un zumbido (beep).

BGET$ número de canal.–Obtiene un carácter de un fichero cuyo número de canal es el especificado.

BLOCK READ.–Como READ para bloques de 512 bytes.

BLOCK WRITE.–Como WRITE para bloques de 512 bytes.

BLOCK WRITE FILE.–Como WRITE FILE para bloques de 512 bytes.

BOLD nombre de programa.–Carga un programa en binario desde la cinta del casete.

BP.–Ver BPUT$.

BPUT$ número de canal, cadena de caracteres.–Graba el byte menos significativo de la cadena de caracteres en el fichero/canal especificado.

NOTAS DE TRADUCCION

BREAK línea número-1, línea número-2, ...—Provoca la parada de la ejecución del programa en las líneas especificadas; la ejecución se rearranca con el comando/sentencia CONTINUE.

BREAK OFF línea número-1, línea número-2, ...—Cancela una sentencia BREAK ON previa, para todos o para determinados bloques de programa.

BREAK ON.—Ver BREAK.

BRIGHTNESS lista de parámetros.—Establece el brillo y los niveles de enfoque en la pantalla de gráficos.

BRK.—Ver BREAK.

BSAVE.—Almacena el programa actual, en modo binario, en la cinta del casete.

BUF.—Utilizado con INPUT; devuelve un valor describiendo el tipo de datos del siguiente elemento del área del buffer de entrada/salida (1-real, 2-cadena, 4-vacío, 5-entero, 6-largo, 7-complejo).

BUFFER número de fichero.—Asigna un área de 256 bytes para lectura/escritura del fichero especificado.

BUFSIZ (canal-N).—Devuelve el tamaño, en bytes, del buffer de entrada/salida del canal N.

BYE.—Utilizado en los grandes sistemas de ordenadores con multiprogramación para salir (de Goodbye = adiós) del BASIC y en sistemas más pequeños para entrar en el DOS (Disk Operating System) o para cerrar el sistema (log-off).

NOTAS DE TRADUCCION

C.—Ver CONT o COLOUR.

CA.—Ver CALL.

CALL nombre de subprograma, (lista de parámetros).—Llama a un programa y comienza la ejecución; la lista de parámetros contiene los nombres de las variables requeridas por el subprograma; normalmente están especificadas en el área denominada COMMON.

CALL dirección de memoria.—Bifurca a una rutina en lenguaje ensamblador ubicada en la posición de memoria especificada.

CATALOG "nombre de fichero, especif. de volumen", "tipo de fichero".—Proporciona el nombre de fichero, el tipo y las especificaciones físicas de cada uno de los ficheros de usuario contenidos en un dispositivo periférico.

CCONT nombre de fichero-A, nombre de fichero-B.—Añade el fichero B al final del fichero A dando lugar a un fichero continuo.

CDBL (variable numérica).—Convierte una variable numérica de simple precisión a doble precisión; la conversión es temporal y permanece solamente para la sentencia CDBL.

CEI (expresión numérica).—Devuelve el "techo" (ceiling), es decir, el menor entero mayor o igual que la expresión especificada.

CH "cadena de caracteres".—Ver ASCII.

NOTAS DE TRADUCCION

CH.—Ver CHAIN.

CHAIN nombre de programa, número de línea WITH fichero número-1, fichero número-2, ...—Llama, compila (si es necesario) y ejecuta el programa especificado a partir del número de línea (opcional) indicado, permitiendo acceder a los ficheros relacionados.

CHANGE cadena de caracteres TO cadena numérica.—Cambia una cadena de caracteres a numérica y viceversa.

CHAR.—Ver CHR$.

CHAR lista de parámetros.—Cambia o imprime una relación de las asignaciones del dispositivo actual.

**CHAR$.**—Ver CHR$.

CHARACTER.—Ver CLEAR CHARACTER ARRAYS.

CHARSIZE entero-C, entero-L.—Especifica el número C de caracteres por línea y el número L de líneas por pantalla.

CHECK READ OFF$ número de fichero.—Verifica que la información que se está almacenando en un dispositivo de almacenamiento masivo es idéntica a la que hay en memoria leyendo dicha información desde el dispositivo a memoria y comparándola.

CHR.—Ver CHR$.

NOTAS DE TRADUCCION

CHR$ (expresión numérica).—Retorna el carácter tipo cadena cuya representación decimal ASCII es la expresión numérica evaluada (0-255).

CINT (expresión numérica).—Convierte una expresión numérica en un entero; la conversión es temporal y permanece solamente durante la sentencia CINT.

CIRCLE (fila X, columna Y) R, C, A, P, F.—Dibuja un círculo centrado en la posición X, Y de pantalla (de gráficos) y de "Radio" y "Color" especificados; el círculo es "alto" o "gordo" si la "Altura" es +/—; "Principio" y "Final" son parámetros para dibujar arcos en lugar de círculos completos.

CISIZE.—Especifica que el TAMAÑO DEL INTERVALO DE CONTROL (CONTROL INTERVAL SIZE) de un fichero se cree igual al tamaño del bloque más la información de cabecera del fichero.

CL.—Ver CLEAR.

CLEAR N.—Restaura todas las variables numéricas a cero y todas las variables tipo cadena a nulos; opcionalmente asigna N bytes para almacenamiento de cadenas.

CLEAR CHARACTER ARRAYS array-1, array-2, ...—Limpia todos los arrays tipo carácter del programa (o los especificados).

CLEAR CHARACTER SCALARS cadena-1, cadena-2, ...—Limpia todas las variables tipo cadena del programa (o las especificadas).

CLEAR NUMERIC ARRAYS array-1, array-2, ...—Limpia todos los arrays numéricos del programa (o los especificados) poniéndolos a cero, en formato entero o en coma flotante dependiendo del tipo de datos.

NOTAS DE TRADUCCION

CLEAR NUMERIC SCALARS.—Limpia todas las variables numéricas del programa (o las especificadas) poniéndolas a cero en formato entero o en coma flotante dependiendo del tipo de datos.

CLI.—Ver MONITOR.

CLG.—Limpia la pantalla de gráficos.

CLK.—Ver CLK$.

CLK$.—Devuelve la hora actual en formato de 24 horas.

CLO.—Ver CLOSE$.

CLOAD.—Carga un programa desde una cinta de casete.

CLOADM nombre de programa, entero-A.—Lee desde una cinta de casete el programa indicado en lenguaje máquina y lo carga en memoria, en la dirección (opcional) de comienzo A.

CLOG.—Ver LGT.

CLOSE nombre de fichero/array virtual.—Termina la entrada/salida entre el programa actual y el dispositivo periférico o el array virtual.

CLOSE$ número de canal.—Como CLOSE para canales físicos.

CLOSE FILE.—Ver CLOSE.

NOTAS DE TRADUCCION

CLR.—Borra todas las referencias almacenadas a variables, arrays, funciones, direcciones de retorno de los GOSUB y FOR-NEXT y contadores de bucles.

CLRDOT fila-X, columna-Y.—Limpia la posición X, Y de la pantalla de gráficos.

CLS.—Limpia la pantalla de gráficos y mueve el cursor a la esquina superior izquierda.

CLOSTERSIZE.—Especifica el número de bloques que se tomará como unidad de lectura de un fichero de datos.

CMD.—Ver MONITOR.

CMP% (cadena-A, cadena-B).—Como COMP% ignorando las diferencias entre mayúsculas y minúsculas.

CNJ (expresión numérica).—Devuelve el conjugado de la expresión numérica, invirtiendo el signo de la parte imaginaria.

CO.—Ver COL.

COL (array).—Devuelve el número de la columna actual del array especificado.

COLOR (=) código de color/código de primer plano, código de fondo.—Fija los colores de la pantalla de gráficos. SETCOLOR crea una lista con los códigos de los colores disponibles.

COLOUR.—Ver COLOR.

NOTAS DE TRADUCCION

COM variable-1, variable-2, ...—Dimensiona y reserva espacio de memoria para variables, simples, arrays y números de ficheros de datos en un área de memoria "común" a la que se puede acceder desde 2 programas o más.

COMMON.—Ver COM.

COMP% (cadena numérica-A, cadena numérica-B).—Compara A y B y devuelve −1 si A es menor que B, 0 si A es igual que B y +1 si A es mayor que B.

COMPILE nombre de fichero.—Compila o traduce el programa cargado en memoria a código objeto e información de usuario y la almacena en el fichero especificado; los códigos de opción indican al compilador si debe obtener un listado impreso, referencias cruzadas y similares.

COMPLEX variable-1, variable-2, ...—Declara que las variables relacionadas son COMPLEJAS.

CON—Ver MAT o CON.

CON.—Ver CONTINUE.

CONT.—Ver CONTINUE.

CONTINUE número de línea.—Reanuda la ejecución del programa actual, opcionalmente a partir del número de línea especificado.

CONVERT.—Convierte valores numéricos en caracteres (y viceversa); si no se puede efectuar la conversión el programa bifurca a la línea de programa especificada.

NOTAS DE TRADUCCION

COPY nombre de fichero-A TO nombre de fichero-B.—Copia un fichero de un medio a otro.

COS (expresión numérica).—Devuelve el conseno del ángulo representado por la expresión numérica (en radianes, no grados).

COSD.—Ver COS.

COSG.—Ver COS.

COSH.—Ver CSH.

COT (expresión numérica).—Retorna la cotangente del ángulo representado por la expresión numérica (en radianes, no grados).

COU.—Ver COUNT.

COUNT entero.—Utilizado con PUT para escribir registros cuyo número de caracteres es menor que el de un registro normal.

COUNT.—Retorna el número de caracteres enviados en ese momento a la pantalla o a la impresora.

CPI.—Ver PI.

CPU (variable).—Devuelve el tiempo de CPU (Unidad Central de Proceso) de ejecución del programa actual, en milisegundos.

NOTAS DE TRADUCCION

CPX (expresión numérica-A, expresión numérica-B).—Calcula el número complejo A + B.

CRAND nombre de fichero.— Crea un fichero random (con organización al azar).

CREATE especif. fichero, número de registros definidos, longitud del registro.—Configura un fichero de datos de tamaño especificado y coloca una marca de EOF (End Of File = Fin de fichero) en cada palabra de todos los registros.

CRM$ (variable tipo cadena).—Guarda cada grupo de tres caracteres tipo cadena (0-9, A-Z y .,- espacio) en 2 bytes.

CSAVE.—Almacena el programa actual en cinta de casete.

CSAVEM.—Almacena el programa actual en lenguaje máquina en cinta de casete.

CSH (valor numérico).—Calcula el coseno hiperbólico del valor numérico especificado.

CSNG expresión numérica.—Converte la expresión numérica en un número en precisión sencilla, la conversión es temporal y permanece para la sentencia CSNG solamente.

CUF$.—Devuelve el nombre del fichero actual de trabajo.

CUR (fila-X, columna-Y).—Utilizado con PRINT; sitúa un carácter en la posición especificada (X, Y) de la pantalla de gráficos.

NOTAS DE TRADUCCION

CURKEY.—Devuelve el número de tecla de función especial que se ha pulsado en el teclado provocando una bifurcación a la rutina o programa actual.

CURSOR caracteres de control.—Controla el cursor y los campos de caracteres a visualizar utilizando los caracteres de control especificados para la optimización de la visualización, protección de líneas y definición de los campos.

CURSOR (X, Y).—En la pantalla de gráficos, mueve el cursor a la posición fila X, columna Y.

CVTXX (variable).—Convierte una variable de tipo cadena a numérica y viceversa dependiendo del valor de XX (%$ = entera a carácter, F$ = coma flotante a carácter, $% = carácter a entera, $F = carácter a coma flotante y $$ = cadena a cadena, con posibilidad de acción sobre los bits de paridad excluyendo los espacios y tabuladores, conversión de mayúsculas/minúsculas y similares).

CVX (cadena-A).—Devuelve los datos contenidos en la cadena-A convertidos a doble o simple precisión o a enteros dependiendo del valor de X = D, I, S y la longitud A = 8, 2 y 4 respectivamente.

NOTAS DE TRADUCCION

D.—Ver DELETE.

D.—Ver DATA.

DASH.—Rellena la pantalla de gráficos con líneas de "guiones" ("dashes").

DAT.—Ver DATA.

DAT$ (entero-1, entero-2).—Devuelve la cadena con la fecha y hora actual en distintos formatos dependiendo del valor de los enteros.

DATA valor o texto, valor o texto, ...—Proporciona valores numéricos o textos con o sin comillas a partir de los cuales la sentencia READ obtiene valores para asignarlos a variables numéricas o de tipo cadena.

DATE.—Devuelve la fecha actual en formato AADDD.

DATE$.—Devuelve la fecha actual con formato AAMMDD.

DATE (entero-1, entero-2).—Ver DAT$.

DAY.—Devuelve el día del mes actual (1-31).

DEB$ (variable tipo cadena).—Devuelve una cadena igual a la especificada pero sin los espacios de cabecera y cola.

DEBUG.—Ver TRACE.

NOTAS DE TRADUCCION

DECLARE ALL INTEGER.—Como DECLARE INTEGER para TODOS los arrays y variables enteros.

DECLARE ALL REAL.—Como DECLARE REAL para TODOS los arrays/variables numéricos.

DECLARE INTEGER.—Ver INTEGER.

DECLARE REAL.—Ver REAL.

DEF.—Ver DEF.

DEF FN X (variable-A, variable-B, ...) = expresión numérica.—permite definir una FUNCION X de manera que pueda ser utilizada exactamente igual que las funciones estándares del sistema; X es cualquier carácter comprendido entre A y Z.

DEFFN.—Ver DEF FN.

DEFAULT ON/OFF.—ON impide que el programa se pare debido a errores aritméticos; OFF permite la parada del programa cuando existen errores aritmético.

DEFDBL D.—Declara como variables de doble precisión a todas aquéllas cuyo nombre comienza por la letra especificada o por el rango especificado.

DEFINIT I.—Declara como variables enteras a todas aquéllas cuyo nombre comienza por la letra especificada o por el rango especificado.

NOTAS DE TRADUCCION

DEFSNG S.—Declara como variables de precisión simple a todas aquéllas cuyo nombre comienza por la letra especificada o por el rango especificado.

DEFSTR V.—Declara como variables de tipo cadena a todas aquéllas cuyo nombre comienza por la letra especificada o por el rango especificado.

DEFURS n = dirección.—Define el punto de entrada para la función USR (n = 0 a 9).

DEG.—Fija el modo GRADOS (DEGREE) para los resultados y argumentos de las funciones trigonométricas en lugar de RADIANES.

DEGREE.—Ver DEG.

DEL (cadena-A, cadena-B, entero-N).—Devuelve una cadena con la primera, opcionalmente la ené-sima, ocurrencia no superpuesta de B dentro de A, borrada.

DEL.—Ver DELETE número de fichero o DELETE número de línea.

DELAY = expresión numérica.—RETARDA la ejecución del programa actual cargado en memoria durante el número de milisegundos especificado.

DELETE número de línea-1, número de línea-2.—Borra las líneas de programa correspondientes a los números especificados, las que se encuentran dentro del rango especificado o desde el comienzo del programa actual hasta la línea especificada inclusive.

NOTAS DE TRADUCCION

DELETE número de fichero.—Borra del fichero indexado especificado, el registro que se leyó en el último GET.

DELFN nombre de función TO END.—Borra de memoria la función/subprograma definida por el usuario.

DELIMIT número de fichero, "cadena".—Especifica un carácter, o una cadena de caracteres, como delimitador o separador de los elementos de los datos de entrada del fichero especificado en lugar de la coma (carácter normal); si no se indica número de fichero se asume entrada desde el terminal.

DELSUB nombre del subprograma TO END.—Borra de memoria el subprograma especificado.

DET (array).—Devuelve el determinante de una matriz cuadrada.

DIF$ (cadena numérica-A, cadena numérica-B).—Calcula la diferencia entre las cadenas A y B.

DIGITS entero-N, entero-D.—Especifica el número máximo N de dígitos, incluyendo el número D de decimales, para los valores numéricos que se pueden imprimir con las sentencias PRINT.

DIM cadena (entero)/array (índice, entero).—Establece las dimensiones o el número de elementos de las cadenas o arrays numéricos; los elementos se ponen a ceros o nulos; si no se utiliza DIM se asume dimensión = 10.

DIM$.—Como DIM excepto para array virtual.

DISABLE.—Inhabilita una declarativa de interrupción ON KEY.

NOTAS DE TRADUCCION

DISK.—Imprime el número de bloques de 256 palabras utilizados y disponibles en la partición del directorio actual.

DISP lista de representación.—Saca a pantalla los elementos de la lista.

DISP USING cadena.—Como PRINT USING, pero en este caso la salida se obtiene en la pantalla.

DIV variable numérica-A DIV variable numérica-B.—Devuelve el entero resultante de A/B.

DLOADM programa, código de velocidad de línea.—Carga un programa en lenguaje máquina desde otro ordenador mediante conexión telefónica a la velocidad especificada de transmisión de los datos.

DMS variable.—Convierte grados decimales en grados, minutos y segundos; la conversión es temporal y permanece solamente para la sentencia DMS.

DO sentencia-A, sentencia-B, ...—Utilizado junto con IF, THEN y ELSE para ejecutar una serie de sentencias.

DOEND.—Completa una serie de sentencias DO.

DOOR LOCK especificación de volumen.—Bloquea la puerta de la unidad de disquete especificada.

DOOR UNLOCK especificación de volumen.—Desbloquea la puerta de todos los dispositivos bloqueados o de la unidad (especif. volumen) de disquete indicada.

NOTAS DE TRADUCCION

DOS.—Salida del BASIC y entrada al Disk Operating System.

DOT (fila-X, columna-Y).—Comprueba el estado ON/OFF de la posición X, Y de la pantalla de gráficos.

DOUBLE.—Ver D.

DR.—Ver DRAW.

DRAW figura N-, AT fila-X, columna-Y.—Dibuja la figura N a partir de una lista pre-codificada desde la posición X, Y, de la pantalla de gráficos.

DRAW fila-X, columna-Y.—Dibuja, en la pantalla de gráficos, una línea desde la posición actual hasta la posición X, Y utilizando el color de fondo actual.

DRAW (fila-A, columna-B).—Dibuja, en la pantalla de gráficos, una línea desde la posición actual hasta la posición (X + A, Y + B).

DRAWTO.—Ver DRAW.

DROUND (expresión numérica, entero-N).—Devuelve la expresión numérica evaluada y redondeada al número N de dígitos significativos.

DSP variable.—Visualiza el valor de una variable y el número de línea de programa asociado a dicha variable.

NOTAS DE TRADUCCION

DUMP nombre de fichero.—Visualiza el contenido de un fichero en caracteres imprimibles.

DYR.—Devuelve el día actual del año (1-366).

NOTAS DE TRADUCCION

E.—Indica simple precisión en valores numéricos.

E. .—Ver ENDPROC o EDIT.

EDIT "diálogo", variable tipo cadena.—Visualiza, después del carácter de diálogo (prompt), el valor actual de la variable tipo cadena especificada y espera hasta que el operador la haya editado.

EDIT.—Provoca la visualización en pantalla de la última línea de programa para permitir su edición.

EDIT KEY$ SFK número.—Se utiliza como ayuda en la entrada de datos desde el teclado; asigna a una tecla especial de función una serie de caracteres que aparecen cada vez que se pulsa dicha tecla.

EL.—Ver ELSE.

ELSE.—Utilizado con IF.

ENABLE.—Habilita aquellas declarativas de interrupciones ON KEY que hubiesen sido previamente inhabilitadas por DISABLE.

END.—Termina la ejecución del programa que hay en memoria.

ENDPROC.—Indica el final de un procedimiento.

ENTER nombre de fichero, variable-A, variable-B, ...—Permite la entrada de datos desde la unidad de pantalla o desde el fichero indicado (opcional).

NOTAS DE TRADUCCION

ENTER nombre de fichero.—Transfiere y mezcla sentencias BASIC desde el fichero especificado (en cinta o en disco) al programa actual en memoria; los números de línea duplicados se sobreescriben.

ENV.—Ver ENVELOPE.

ENVELOPE lista de parámetros.—Controla el nivel del sonido generado dependiendo del valor de los parámetros (14 en total).

EOF$ número de canal.—Pregunta por fin de fichero y devuelve 0 si se ha encontrado y −1 en caso contrario.

EOR.—OR exclusivo.

EPS.—Devuelve el valor más pequeño tal que 1 + EPS y 1 − EPS no sean iguales a 1.

EQ.—Equivalente a EQUAL TO.

EQUIV.—Ver RENAME.

condición-A EQV condición-B.—Utilizado en sentencias IF; devuelve −1 si A y B son ambas falsa o verdaderas, en caso contrario devuelve 0.

ERASE número de línea-1, número de línea-2.—Borra todos los bloques, o sólo los especificados, del programa cargado en memoria.

ERL.—Ver ERRL.

NOTAS DE TRADUCCION

**ERM$.**—Ver ERRM$.

ERR.—Ver ERRN.

ERR. .—Ver ERROR.

ERRL.—Devuelve el número de la línea de programa en la que se ha producido el último error de ejecución.

ERRM$.—Devuelve el mensaje de error del último error de ejecución del programa.

ERRN.—Devuelve el número de mensaje del último error de ejecución del programa.

ERROR.—Simula condiciones de error para pruebas.

ESC

EU.—Ver EVAL.

EVAL (expresión numérica).—Obtiene el valor de la expresión numérica.

EXAM.—Ver PEEK.

EXCHANGE variable, variable
cadena, cadena.—Intercambia los valores de las variables especificadas o de los elementos de las cadenas.

NOTAS DE TRADUCCION

EXEC dirección de memoria.—Ejecuta una rutina en lenguaje máquina ubicada en la dirección de memoria especificada.

EXEC.—Ver MONITOR.

EXIT.—Utilizado para SALIR de un bucle FOR-NEXT antes de la terminación normal; el valor del contador STEP para el bucle se mantiene.

EXP (expresión numérica).—Devuelve el valor del número de Neper (e) elevado a la expresión numérica evaluada.

EXT$ número de canal/nombre de fichero.—Devuelve el tamaño en bytes del fichero especificaco.

EXTEND.—Identifica una versión ampliada del BASIC.

NOTAS DE TRADUCCION

F.—Ver FOR.

FA.—Ver FALSE.

FALSE condición.—Devuelve el valor 0 si la condición no es verdadera.

FDIM.—Versión de DIM para punto flotante.

FETCH número de línea.—Visualiza la línea de programa especificada.

FGET.—Lee, desde un fichero, números en punto flotante.

FIELD cadena-A, cadena-B, cadena-C, ...—Relaciona los nombres de las cadenas con la totalidad o con una parte del buffer de entrada/salida.

FIELD número de buffer, longitud-1 AS variable-1, longitud-2 AS variable-2, ...—Organiza un área (256 bytes) para un fichero RANDOM en campos cuyas longitudes son las especificadas.

FIF.—Versión especial del IF para operaciones de punto flotante.

FILE nombre de fichero-1, nombre de fichero-2, ...—Deja disponibles los ficheros para la entrada/salida de datos y les asigna un número, empezando en 1, utilizado para referenciar a cada fichero desde el programa; se pueden reservar los números para posteriores ASSIGN (ASIGNACIONES) de ficheros mediante un asterisco en la lista.

FILE número de canal, nombre de fichero/*.—Empareja los números de canal con nombres de ficheros y habilita el fichero especificado para la entrada/salida de datos; el * cancela la asociación entre el canal y el fichero.

NOTAS DE TRADUCCION

FILE CREATE número de canal; nombre de fichero/*.—Como FILE excepto que el fichero se crea en el área de almacenamiento para ficheros de usuarios.

FILES.—Imprime los nombres de los ficheros abiertos por el programa actual.

FILESIZE entero.—Especifica el número de registros de un fichero.

FILL.—Ver POKE.

FILL$ (expresión numérica, entero-M).—Devuelve una cadena de M caracteres siendo cada uno de ellos el valor decimal ASCII de la expresión numérica evaluada.

FIN.—Ver FILE.

FIND nombre de fichero.—Fija el puntero de registro de fichero al comienzo del fichero especificado.

FINPUT.—Versión especial de INPUT para números en punto flotante.

FIRST.—Referencia la PRIMERA línea del programa actual.

FIX (expresión numérica).—Desprecia los dígitos que se encuentran a la derecha de coma/punto decimal sin hacer redondeo.

FIXED número de dígitos.—Establece el punto fijo para los valores numéricos y especifica que el número de dígitos a la derecha del punto decimal variará de 1 a 12.

NOTAS DE TRADUCCION

FIXED.—Ver ORGANIZATION.

FLASH.—Utilizado para visualizar alternativamente en pantalla caracteres blancos en fondo negro y viceversa.

FLOAT número de dígitos.—Establece el punto/coma flotante para la salida de los valores numéricos y especifica que el número de dígitos a la derecha del punto decimal variará de 1 a 11.

FLOW.—Ver TRACE.

FLT expresión numérica.—Convierte en punto flotante la expresión numérica especificada.

FMT (expresión numérica, formato codificado).—Utilizado con PRINT: edita y formatea valores numéricos tales como supresión de ceros, signos especiales y la coma/punto decimal.

FN.—Ver DEFFN.

FNEND.—Ultima línea (opcional) de una función/subprograma.

FONT códigos del juego de caracteres.—Selecciona los juegos de caracteres para la impresora a partir de una lista precodificada.

FOR condición TO entero.—Asigna rangos de valores a determinadas variables para controlar los bucles del programa; normalmente se utiliza NEXT para controlar el incremento en 1 mientras se cumple la condición.

NOTAS DE TRADUCCION

FOUT nombre de fichero.—Permite la escritura (output) en un fichero random y asigna un número de fichero.

FP.—Ver FRACT.

FPRINT variable en punto flotante.—Versión especial de PRINT para números en punto flotante.

FPUT variable en punto flotante.—Escribe en un fichero, números en punto flotante.

FRA.—Ver FRACT.

FRAC.—Ver FRACT.

FRACT (expresión numérica).—Retorna la parte fraccionaria de la expresión evaluada/número real.

FRE.—Devuelve la cantidad de espacio que hay disponible para almacenamiento de cadenas.

FREE.—Ver FRE.

FUNTIL.—Versión especial de UNTIL.

FUZZ entero-N/entorno del cero.—Establece el número N de dígitos que se utilizarán para la comparación entre variables (los N primeros); también define que los números comprendidos en un entorno determinado del cero pueden ser considerados iguales a cero.

NOTAS DE TRADUCCION

G.–Ver GOTO o GCOL.

**GAP$.**–Ver SPACE$.

GCOL modalidad-M, entero-N.–Dependiendo de N fija el color de primer plano (N de 0 a 127) o de fondo (N de 128 a 255) para los siguientes gráficos; la modalidad determina si se trabajará con el color seleccionado directamente o si se combinará con el color actual.

GE.–Equivalente a GREATER THAN OR EQUAL TO.

GET.–Espera indefinidamente que se pulse una tecla y devuelve su representación decimal ASCII.

GET nombre de programa, número de línea-1, número de línea-2.–Trae a memoria un programa, o cualquier fichero de datos tipo cadena constituido por sentencias BASIC válidas, salvado con una sentencia SAVE; opcionalmente se puede renumerar el programa a partir del número de línea-1 y la ejecución comienza en el número de línea-2.

GET número de fichero.–Lee el siguiente registro lógico del fichero especificado.

GET número de fichero RECORD posición.–Lee del fichero especificado, el registro (RECORD) indicado por la "posición".

GET número de fichero, KEY cadena-A.–Lee del fichero especificado, el registro identificado por la clave (KEY) dada por la cadena-A.

NOTAS DE TRADUCCION

GET número de buffer, número de registro.—Lee un registro especificado (o el siguiente registro) desde un fichero random en disco y lo carga en el área del buffer del fichero especificado.

GET$.—Como GET pero devuelve el carácter tipo cadena que corresponde a la tecla pulsada.

GET (fila-X, columna-Y) - (fila-A, columna-B), array-G.—Almacena el rectángulo (gráfico) definido por las posiciones de pantalla X, Y (esquina superior izquierda) y A, B (esquina inferior derecha), en el array-G.

GIN (fila-X, columna-Y).—Fija las coordenadas actuales del cursor en X, Y.

GO número de línea.—Ver RESUME.

GOS.—Ver GOSUB.

GOSUB X OF subrutina-1, subrutina-2.—Bifurca a la subrutina-1 si X = 1 y a la subrutina-2 si X = 2; retorna a la línea de programa siguiente a la sentencia GOSUB cuando se finaliza la ejecución de la subrutina.

GOSUB número de línea.—Bifurca al número de línea de programa y retorna a la línea de programa que está a continuación del GOSUB cuando se haya ejecutado la subrutina.

GOTO número de línea.—Bifurca a la línea de programa especificada.

GOTO X of número de línea-1, número de línea 2, ...—Bifurca a la línea-1 especificada si X = 1, a la línea-2 si X = 2, etc.

NOTAS DE TRADUCCION

GPOS FILE (nombre de fichero), variable.—Determina la posición del punt del fichero actual y asigna el valor a la variable especificada.

GR.—Establece la modalidad de gráficos.

GR.—Establece la modalidad de gráficos en baja resolución.

GRAD.—Todos los resultados y argumentos de las funciones trigonométric se darán en GRADOS (1 grado = 1/400 de un círculo).

GRAPH

GRAPHICS código de rango.—Selecciona el número de rango de las líneas d gráficos que se va a visualizar en pantalla, a partir de una lista pre-codifica da en memoria.

GT.—Equivalente a GREATER THAN.

NOTAS DE TRADUCCION

H.—Ver HIMEM.

HCOLOUR = código de color.—Establece el color para gráficos en alta resolución.

HELP "expresión BASIC".—Visualiza información sobre cada sentencia, mandato, función y operador BASIC.

HEX$ (expresión numérica).—Devuelve el valor hexadecimal de la expresión numérica evaluada.

HGR entero-P.—Limpia la pantalla de gráficos y visualiza la página P de gráficos.

HIMEM.—Devuelve la dirección del byte por encima del que está utilizando actualmente el intérprete BASIC.

HLIN columnas X, Y AT fila-A.—Visualiza una línea horizontal en la fila A y desde la columna X a la Y de la pantalla de gráficos.

HOLE.—Devuelve el mayor número de registros libres contiguos del dispositivo de almacenamiento masivo asumido por defecto.

HOME.—Ver CLS.

HPLOT columna-X, fila-Y TO columna-A, fila B, TO, ...—Pone a la posición X, Y de la pantalla de gráficos con el color que se especificó en el último

---

NOTAS DE TRADUCCION

HCOLOUR; opcionalmente se puede dibujar una línea coloreada hasta A, B y después hasta...; si no se especifica X, Y se asume el último punto referenciado.

HTAB columna-X.—Coloca el cursor en la posición especificada.

NOTAS DE TRADUCCION

I.—Ver INPUT.

I. .—Ver INPUT o INT.

IDN.—Ver MAT.

IF expresión/condición OR ... THEN sentencia ELSE sentencia.—Proporciona al programa caminos lógicos o bifurcaciones condicionales.

IF END nombre de fichero, sentencia.—Alternativa para ON END.

IMAGE formato de imagen.—Usado con PRINT USING o DISP USING; especifica el control del formato de las salidas a pantalla o impresora incluyendo campos para edición (supresión de ceros y similares).

IMG (expresión numérica).—Devuelve la parte imaginaria de la expresión numérica especificada.

IMMEDIATE.—Inicializa la modalidad INMEDIATA para que las líneas de programa se ejecuten nada más ser tecleadas.

cadena-A IMP cadena-B.—Implicación lógica usada con IF. Pone el indicador a 1 si A es verdadero y B falso; en los demás casos lo pone a −1.

IN.—Ver INPUT.

INCH.—Ver INKEY$.

NOTAS DE TRADUCCION

INCHAR.—Ver INKEY$.

INCSIZE.—Especifica el TAMAÑO DE INCREMENTO DE FICHERO (FILE INCREMENT SIZE), que es la cantidad de espacio adicional que se le asignará cuando el espacio existente esté lleno.

INDEX (cadena-A, cadena-B).—Devuelve la posición del primer caracter de la segunda cadena que ocurre en la primera; si la segunda cadena no constituye parte de la primera devuelve un 0.

INDEXED.—Ver ORGANIZATION.

INF.—Devuelve el mayor valor numérico permitido (Infinito).

INIT nombre de fichero, número de disco, número de slot, número de volumen.—Inicializa una unidad de disco almacenando el programa actual cargado en memoria y asignando un número de serie de volumen; el programa almacenado se ejecuta en forma automática cada vez que se inicializa el disco.

INIT.—Inicializa todos los parámetros del programa con los valores asumidos por defecto.

INK.—Ver INKEY$.

INKEY (entero).—Espera durante el tiempo especificado a que se pulse una tecla devolviendo el código —1 si no se ha pulsado ninguna tecla y 0 en caso contrario.

INKEY$.—Monitoriza el teclado durante un lapso corto de tiempo y visualiza/devuelve la tecla pulsada (cadena nula si no se pulsa ninguna tecla).

NOTAS DE TRADUCCION

INP (entero-P).—Devuelve el valor de la representación decimal ASCII del punto P de la pantalla de gráficos.

INPUT "diálogo" nombre de variable, nombre de variable, ...—Suspende la ejecución del programa, visualiza el carácter de diálogo (prompt) o un mensaje y permite la asignación de valores a las variables especificadas, desde el teclado.

INPUT LINE número de fichero, variable tipo cadena.—Versión especial de INPUT para asignar una línea de caracteres del teclado desde el terminal o desde un fichero (opcional) a una variable tipo cadena.

INPUT$ (entero).—Permite la entrada de un número exacto de caracteres desde el teclado sin visualizarlos en pantalla y sin necesidad de pulsar la tecla ENTER o RETURN.

INPUT1 variable.—Similar a INPUT pero inhabilita/ignora los mandatos de retorno de carro y alimentación de línea de manera que los datos de entrada se visualizan en la misma línea que la petición de entrada.

INS.—Ver INSTR.

INSTR (entero-N, cadena-A, cadena-B).—Busca la existencia de la cadena B en la cadena A a partir de la posición N; devuelve el valor 0 si no se encuentra, en caso contrario devuelve la posición de A a partir de la cual se encuentra B.

INSTRING.—Ver INSTR.

INT.—Especifica INTEGER BASIC.

NOTAS DE TRADUCCION

INT (expresión numérica).—Devuelve el valor del mayor entero que es menor o igual a la expresión evaluada; desprecia los dígitos que hay a la derecha del punto decimal y redondea a entero por defecto.

INTEGER variable numérica/array.—Dimensiona y reserva espacio para almacenar variables de tipo entero y arrays.

ITM (número de fichero).—Utilizado con punteros de ficheros para identificar el número de elementos que hay entre el registro actual y el puntero.

INV.—Ver MAT.

INVERSE.—Cambia los colores de fondo y de los caracteres en pantalla de negro a blanco y viceversa.

INVOKE "nombre de programa", número de línea.—Suspende el programa actual, carga y ejecuta el programa especificado a partir de la línea (opcional) especificada; la ejecución del programa llamador continúa cuando se ha completado la del programa llamado.

INVOL

IP (expresión numérica).—Devuelve la parte entera de la expresión evaluada o de un número real.

NOTAS DE TRADUCCION

JOYSTYK (entero).—Devuelve un valor de 0 a 63 indicando la posición del mando de juego (joystick) especificado (entero = 0, 1, 2, 3).

NOTAS DE TRADUCCION

KADD descriptor de fichero, buffer, clave, número de registro.—Añade al fichero descrito, la información del buffer que corresponde al registro/clave especificado, devolviendo el número de registro = 0 en caso de error.

KDEL descriptor de fichero, buffer, clave, número de registro.—Borra del fichero descrito, el registro correspondiente al número de registro/clave especificado, devolviendo el número de registro = 0 en caso de error.

KEY.—Utilizado para pasar al usuario de modo de entrada TAPE a modo KEYBOARD (de CINTA a TECLADO).

KEY$ (entero).—Acepta un único carácter desde el teclado sin visualizarlo; el entero es 0 para el teclado, 1 para teclas de la mano derecha y 2 para teclas de la mano izquierda, en el control de juegos.

KEY$ (número de fichero).—Devuelve una cadena conteniendo la CLAVE del último registro accedido del fichero especificado.

KEYOFFSET entero.—Especifica la posición dentro del registro de la CLAVE que identifica unívocamente a un registro en un fichero Indexado.

KEYSIZE entero.—Especifica el tamaño de bytes del campo de la clave para un fichero indexado.

KILL nombre de fichero.—Borra un fichero del disco.

KFIND descriptor de fichero, buffer, clave, número de registro.—Localiza la clave especificada en el fichero indicado y devuelve el número de registro = = puntero de registro.

NOTAS DE TRADUCCION

KNEXT descriptor de fichero, buffer, clave, número de registro.—Se utiliza con KFIND para empezar a leer datos a partir del registro siguiente al especificado por el puntero hallado en KFIND.

NOTAS DE TRADUCCION

L.—Ver LEN o LIST.

LAST.—Referencia la última línea del programa actual en memoria.

LDSIP lista a visualizar.—Visualiza en pantalla la lista de elementos en la siguiente línea desprotegida.

LE.—Equivale a MENOR QUE o IGUAL A.

LE.—Ver LEFT$.

LEFT (cadena-A, entero-N).—Devuelve los N caracteres de la izquierda de la cadena-A.

LEFT$.—Ver LEFT.

LEN (variable tipo cadena).—Devuelve un valor igual al número de caracteres de la variable especificada.

LENGTH.—Devuelve el tamaño medido en K(1024) bytes del programa actual y el tamaño máximo de la memoria disponible para el usuario.

LENTER variable tipo cadena.—Lee el contenido de la línea actual de pantalla asignándolo a la variable tipo cadena especificada y continúa la ejecución del programa.

LET variable numérica/variable tipo cadena = expresión.—Asigna un valor a una o más variables numéricas o de tipo cadena.

NOTAS DE TRADUCCION

LGT (entero).—Devuelve el logaritmo decimal del entero (positivo) especificado.

LI.—Ver LIST.

LIBRARY programa-1, programa-2, ...—Indica que los programas relacionados deben incluirse para la compilación del programa actual.

LIM.—Especifica el límite de tiempo de ejecución para el programa actual.

LIN (entero).—Utilizado con PRINT y DISP; produce la salida de los caracteres de control de un retorno de carro y el número de saltos de línea especificado; si el entero es negativo no se produce el carácter de control para el retorno de carro.

LIN (cadena-A).—Devuelve el número de caracteres de control de carro que hay en la cadena especificada.

LINE.—Devuelve el número de línea del programa actual.

LINE (fila-X, coluna-Y) - (fila-A, columna-B), PSET/PRESET, B/BF.—Dibuja una línea en la pantalla de gráficos desde la posición X, Y (128, 96 si no se especifica) a la posición A, B usando los colores de primer plano (PSET) o de fondo (PRESET); B dibuja una caja (box); BF dibuja una caja rellena (filled box).

LINE INPUT.—Ver INPUT.

LINE INPUT HASH.—Ver INPUT LINE.

NOTAS DE TRADUCCION

LINK programa, número de línea-1, número de línea-2.—Ver GET programa.

LINPUT "diálogo", variable tipo cadena.—Ver INPUT LINE.

LINPUT número de fichero, cadena.—Ver INPUT LINE.

LIS.—Ver LIST.

LIST número de línea-1, número de línea-2.—Visualiza en pantalla o lista en la impresora, el programa cargado de memoria; opcionalmente sólo las líneas especificadas.

LIST.—Visualiza la última línea del programa actual cargado en memoria.

LIST KEY$ SFK lista de números.—Lista todas las definiciones de ayudas sobre el teclado de las teclas de función especificadas o todas las definiciones.

LISTNH.—Como LIST pero con la opción NO HEADING, es decir, sin líneas de cabecera para el nombre de programa, fecha, etc.

LIST0 entero.—Establece las opciones para formatear un programa LISTED insertando líneas de espacios; el entero varía de 0 a 7.

LLIST.—Lista el programa actual cargado en memoria, en la impresora en lugar de la pantalla.

LN (entero).—Devuelve el logaritmo natural en base "e" del entero positivo especificado.

NOTAS DE TRADUCCION

LO.–Ver LOAD.

LOAD nombre de fichero, número de línea.–Carga en memoria un programa almacenado con STORE y empieza la ejecución en la línea opcional especificada.

LOAD BIN nombre de fichero.–Carga en memoria el fichero binario especificado.

LOAD KEY nombre de fichero.–Carga las definiciones de ayudas sobre el teclado para las teclas de función especiales, desde un fichero almacenado con el mandato STORE KEY.

LOAD SUB nombre de programa, número de línea, incremento para el segmento inicial, último segmento.–Carga uno o más subprogramas desde un fichero de programas.

LOC.–Ver LOCAL.

LOC nombre de fichero.–Devuelve la posición del puntero del fichero especificado indicando cuál es el registro accedido.

LOCAL variable tipo cadena-A, variable tipo cadena-B, ...–Utilizado para almacenar y restaurar el valor de las variables especificadas antes y después de un procedimiento de manera que dichas variables puedan emplearse y tomar distintos valores en el procedimiento.

LOCATE columna-X, fila-Y, variable-N.–Almacena en N la representación decimal ASCII del valor (0-255) de la posición (X, Y) de la pantalla de gráficos.

NOTAS DE TRADUCCION

LOCK nombre de fichero, número de disco, número de slot, número de volumen.—Protege a un fichero de las operaciones de escritura, actualización, borrado y renombrado hasta que se le envíe el mandato UNLOCKED.

LOCK nombre de fichero, variable de espera.—Utilizado para restringir el acceso a ficheros de datos en almacenamiento masivo, desde la consola o desde el programa; el parámetro opcional de espera establece el período de espera si el fichero ya está bloqueado.

LOF nombre de fichero/número de canal.—Devuelve la longitud del fichero especificado en número de registros o de líneas que contiene el fichero.

LOG.—Ver LN.

LOG10.—Ver LGT.

LOGE.—Ver LN.

LOM.—Ver LOMEM.

LOMEM.—Opuesto de HIMEM; devuelve la dirección del byte por debajo del utilizado por el intérprete BASIC.

LON.—Longitud máxima permitida para una cadena (4095).

LONG variable-1, variable-2, ...—Declara que las variables serán de tipo LONG.

LOP número de canal.—Devuelve la posición que ocupa dentro del registro el carácter referenciado por el puntero de fichero actual del canal especificado.

NOTAS DE TRADUCCION

LPRINT.–Similar a PRINT pero escribe en la impresora en lugar de la pantalla.

LREAD.–Ver LINPUT.

LREAD FILE.–Como LINPUT desde un disquete.

LSET cadena = "juego de caracteres".–Almacena los datos de una cadena justificados a la izquierda (normalmente en el área del buffer de entrada/salida).

LT.–Equivalente a LESS THAN.

LWC$ cadena.–Devuelve una cadena con todas las letras mayúsculas convertidas a minúsculas.

LWRITE variable tipo cadena.–Envía una cadena al terminal.

LWRITE FILE.–Como LWRITE pero a un fichero organizado secuencialmente o al azar.

NOTAS DE TRADUCCION

M.—Ver MEM. o MID$.

MAN.—Permite la entrada manual de números de líneas de programa; los números de línea se asignan normalmente en forma automática empezando por 10.

MARGIN nombre de fichero, tamaño de margen.—Establece la longitud de las líneas a imprimir/visualizar para un fichero especificado.

MARK número de fichero.—Establece una marca de registro en cinta para el almacenamiento de un programa o de un fichero de datos.

MASS STORAGE IS especificación de volumen.—Especifica el dispositivo de almacenamiento masivo que se tomará por defecto.

MAT array = CON.—Asigna un valor constante (normalmente 1) a todos los elementos del array.

MAT array = IDN.—Da la matriz identidad para matrices cuadradas poniendo los elementos de la diagonal principal a 1's y el resto a 0's.

MAT array-B = TRN (array-A).—Transpone dos matrices de manera que las filas de A se convierten en las columnas de B y las columnas de A en las filas de B.

MAT array = ZER.—Pone a ceros los elementos de un array.

MAT array-C = array-A + array-B.—Almacena en el array-C los valores de los correspondientes elementos de array-A + array-B; todos los arrays deben tener las mismas dimensiones.

NOTAS DE TRADUCCION

MAT array-B = array-A.—Copia los valores del array-A al array-B.

MAT array-C = array-A – array-B.—Almacena en el array-C los valores de los correspondientes elementos de array-A – array-B; todos los arrays deben tener las mismas dimensiones.

MAT array-C = (entero) * array-A
array-C = array-A * array-B.—Almacena en el array-C los elementos correspondientes al producto de un número entero por el array-A; el producto del array-A por el array-B.

MAT INPUT número de fichero, array-1, array-2, ...—Acepta datos de entrada desde el fichero especificado (si no se indica, se asume desde el terminal) y rellena los arrays especificados fila a fila.

MAT PRINT número de fichero, array.—Imprime el contenido del array, o de determinados elementos del array, en el terminal o en el fichero especificado (opcional).

MAT READ array.—Asigna valores a un array a partir de una sentencia DATA rellenando el array por filas.

MAX (variable-A, variable-B).—Devuelve el valor mayor de una lista de expresiones numéricas.

MAXFILES límite.—Especifica el número máximo de ficheros en disco que pueden estar activos simultáneamente.

MAXSIZE entero.—Especifica el tamaño máximo, en bloques/intervalos de control que puede alcanzar un fichero.

NOTAS DE TRADUCCION

MEM.—Devuelve el número de bytes desprotegidos y disponibles de la memoria.

MEMORY.—Devuelve el número de palabras utilizadas por el programa actual en memoria y el número total de palabras que todavía quedan disponibles (una palabra = dos bytes).

MEMSIZE.—Devuelve el número de bloques de memoria (en Kbytes = 1024 bytes) utilizados para el programa BASIC actual.

MERGE nombre de programa, número de línea, incremento.—Toma líneas de programa de un fichero de datos y las posiciona en memoria con numeración/incremento (opcionales) de las líneas.

MID (variable tipo cadena, entero-N, entero-M).—Devuelve N caracteres a partir del carácter M de la cadena especificada.

MID$.—Ver MID.

MIN (variable-A, variable-B, ...).—Devuelve el valor más pequeño de una lista de variables o expresiones numéricas.

MKX$ (expresión numérica).—Devuelve la expresión numérica convertida en cadena dependiendo de X = D (Doble), I (Entera) o S (Simple); la longitud de la cadena será 8, 2 ó 4 respectivamente.

MO.—Ver MODE.

MOD: expresión númerica-A MOD expresión numérica-B.—Devuelve el resto entero de A/B.

NOTAS DE TRADUCCION

MOD (expresión numérica-X, expresión numérica-Y).—Devuelve el entero X módulo Y, es decir el resto entero de X/Y.

MODE.—Utilizado con OPEN para indicar si el fichero será de entrada o de salida.

MODE.—Establece las combinaciones de color/tamaño de pantalla.

MON.—Devuelve el mes actual (1-12).

MON C/I/O.—Se visualizan en pantalla los mandatos o comandos (C = Command) del disco, los datos de entrada (I = Input) o los datos de salida (O = = Output).

MONITOR.—Realiza la salida del BASIC y entrada al MONITOR o al Disk Operating system.

MOTOR ON/OFF.—Conecta/desconecta el motor de la casete.

MOV.—Ver MOVE.

MOVE columna-X, fila-Y.—Mueve el cursor en la pantalla de gráficos, a la posición X, Y sin dibujar nada.

MPY.—Ver MAT*.

MRG número de canal.—Devuelve la longitud de la línea actual para el canal especificado.

NOTAS DE TRADUCCION

MSG identidad, "texto del mensaje".—Distribuye un mensaje por el sistema.

MSI.—Ver MASS STORAGE IS.

MTDIO número de canal, código de función.—Utilizado para controlar la entrada/salida desde cinta magnética y desde casete.

MTPACK.—Comprueba el estado físico de la cinta de casete y si se precisa algún ajuste.

**MUL$.**—Ver REPEAT$.

NOTAS DE TRADUCCION

N.—Ver NEW o NEXT.

NAME fichero-A AS fichero-B.—Utilizado para renombrar ficheros.

NE.—Equivalente a NOT EQUAL.

NEW.—Borra el programa actual de memoria, pone las variables numéricas a cero y todas las cadenas a nulos.

NEX.—Ver NEXT.

NEXT controlador de bucle.—Utilizado por FOR para definir la última sentencia de un bucle FOR-NEXT.

NO DISP variable.—Cancela la modalidad DISPLAY de la variable especificada.

NOEXTEND.—Indica una versión no ampliada del BASIC.

NOFLOW.—Desactiva la función FLOW.

NOMON.—Cancela o anula el comando MON.

NORMAL.—Cancela todas las operaciones de TRACE.

NORMAL.—Desactiva las modalidades de vídeo INVERSE y FLASH.

NOTAS DE TRADUCCION

NOTE número de fichero, sector-S, byte-B.—Devuelve la dirección física del registro al que se ha accedido y la almacena en S y B como una dirección SECTOR/BYTE.

NO TRACE.—Ver TROFF.

NUM (expresión tipo cadena).—Devuelve la expresión decimal ASCII (0-255) del primer carácter de la cadena.

NUM.—Devuelve el número de filas de entrada para una Matriz.

NUM$ (cadena numérica).—Convierte una cadena numérica en una cadena de caracteres.

NUM1$ (expresión numérica).—Convierte una expresión numérica en una cadena de caracteres.

NUM2.—Devuelve el número de caracteres introducidos en la última fila de entrada para una matriz bi-dimensional.

NUMERIC.—Ver CLEAR NUMERIC ARRAYS/SCALARS.

NOTAS DE TRADUCCION

O.—Ver OLD.

OCC (cadena-A, cadena-B).—Devuelve el número de OCURRENCIAS no superpuestas de B en A; devuelve el valor 0 si B es una cadena nula.

ODD (expresión numérica).—Devuelve el valor 0 si la expresión evaluada es par, 1 si es impar.

OFF.—Cancela cualquier condición ON previa.

OFF END número de fichero.—Cancela cualquier condición ON END activa para el fichero especificado.

OFF ERROR.—Cancela cualquier condición ON ERROR activa.

OFF HALT.—Cancela cualquier condición ON HALT.

OFF KEY número de tecla.—Desactiva la correspondiente sentencia ON KEY.

OLD.—Recupera un programa borrado con NEW.

ON entero GOSUB rutina-1, rutina-2, ...—Provoca la bifurcación a una rutina dependiendo del valor del entero (1, 2, ...).

ON entero GOTO número línea 1, número línea 2, ...—Provoca la bifurcación a una línea de programa dependiendo del valor del entero (1, 2, ...).

NOTAS DE TRADUCCION

ON BREAK GOTO/GOSUB/CALL.—Bifurca a la línea del programa/a la subrutina/al programa llamado, cuando se produce un BREAK.

ON END número de fichero GOTO/GOSUB/CALL.—Efectúa una bifurcación en la lógica del programa cuando se encuentra un FINAL DE FICHERO (EN OF FILE) durante la ejecución de un READ de un fichero de datos.

ON EOF número de fichero GOTO/GOSUB/CALL.—Bifurca a la línea de programa/a la subrutina/al programa llamado cuando se detecta el final del fichero especificado.

ON ERROR GOTO/GOSUB/CALL.—Utilizado para evitar que algunos errores recuperables, de ejecución, puedan parar el programa. Provoca una bifurcación o llamada a la rutina de tratamiento de error cuando éste se produzca.

ON ESC.—Provoca una bifurcación en la lógica del programa actual cuando se pulsa la tecla ESCAPE.

ON HALT GOTO/GOSUB/CALL.—Ejecuta una bifurcación en la lógica del programa cuando se pulsa la tecla HALT.

ON KEY número de tecla GOTO/GOSYB/CALL.—Ejecuta una bifurcación en la lógica del programa cuando se pulsa una tecla de función especial.

OP.—Ver OPENIN.

OPEN nombre de fichero AS entero.—Deja un fichero disponible para la entrada y/o salida de datos bajo el control del programa y asigna un número utilizado para referenciar el fichero.

NOTAS DE TRADUCCION

OPENFILE.—Ver OPEN.

OPENIN nombre de fichero.—Abre el fichero especificado para entrada y actualización, y asigna un número de canal; si el fichero no existe se devuelve el valor 0.

OPENO.—Ver OPENOUT.

OPENOUT nombre de fichero.—Abre el fichero especificado para salida y actualización (si el fichero ya existe).

OPT entero.—Controla el listado de diagnósticos/errores de un listado del programa. El entero puede tomar valores de 0 a 3.

OPTION BASE.—Ver BASE.

OR.—Ver IF, THEN, ELSE.

ORD (variable tipo cadena).—Devuelve la representación decimal ASCII de la variable tipo cadena.

ORGANIZATION.—Utilizado con OPEN; especifica la organización de un fichero como SECUENCIAL, RELATIVA o INDEXADA (SEQUENTIAL, RELATIVE o INDEXED) con longitud de registro FIJA o VARIABLE (FIXED o VARIABLE).

OUT posición-X, entero-Y.—Coloca el valor Y (0-255) en la posición X de pantalla.

NOTAS DE TRADUCCION

P.–Ver PRINT.

P. .–Ver PRINT.

P.A.–Ver PRINT AT.

PA..–Ver PAGE.

PAD$ (cadena-A, entero-L, cadena-B).–Devuelve una cadena de longitud L conteniendo el valor de la cadena-A rellenada por la derecha con el primer carácter de la cadena-B.

PADDLE (entero-N).–Devuelve un valor (1-228) dependiendo del giro del mando de control especificado por N.

PAGE.–Utilizado con PRINT, DISP o LDISP. Envía a la impresora un carácter de alimentación de formulario para colocar el papel al comienzo de la página.

PAGE = expresión numérica.–Establece el margen derecho para el terminal, en la posición especificada por la expresión numérica evaluada.

PAGE expresión numérica-P, expresión numérica-L.–Establece el número de líneas por página (P) y el número de líneas reservadas (L) en la página actual.

PAINT (fila-X, columna-Y), código de color-F, código de color-B.–Pinta la pantalla de gráficos con el color F hasta que encuentre un borde de color B.

NOTAS DE TRADUCCION

PATH cadena.—Utilizado con OPEN para especificar el nombre de directorio para un fichero.

PAUSE.—Suspende la ejecución del programa actual indefinidamente, hasta el CONTINUE.

PCLEAR entero-N.—Reserva N páginas de 1536 bytes de memoria para utilizarlas con GRAPHICS.

PCLS código de color-N.—Limpia la pantalla de gráficos y fija el color N o el color de fondo actual si no se especifica nada.

PCOPY página-N, página-M.—Copia la página N de gráficos a la página M.

PDL (0)/(1).—Establece las unidades de control de juegos a 0 ó 1.

PEEK dirección de memoria.—Devuelve el valor almacenado en la dirección de memoria especificada.

PER canal-N, cadena-A.—Si la cadena-A no contiene alguno de los valores siguientes: —FILE, INPUT, LINPUT, MARGIN, PRINT, READ, READ$, SCRATCH, WRITE, WRITE$ devuelve un valor —1; si contiene FILE y existe un fichero asignado a N devuelve 0.

PI.—Devuelve el valor pi (π).

NOTAS DE TRADUCCION

PIN (dirección de memoria).—Utilizado con PRINT para imprimir el valor decimal contenido en la dirección de memoria especificada.

PIX (expresión numérica).—Función igual a pi** (expresión numérica).

PL.—Ver PLOT.

PLACE$ (variable tipo cadena, P%).—Redondea la variable de tipo cadena a P posiciones decimales.

PLAY "cadena musical".—Genera tonos musicales de acuerdo con una secuencia de parámetros que especifica notas, tiempo, volumen y similares.

PLOT (columna-X, fila-Y).—Activa la condición ON para la posición de pantalla especificada.

PMODE tamaño de malla-G, página-P.—Establece el tamaño de malla/resolución G y especifica la página de gráficos P que se va a utilizar.

PO.—Ver POINT.

POINT (columna-X, fila-Y).—Comprueba la posición X, Y de la pantalla de gráficos y devuelve —1 si ON y 0 si OFF.

POINTER columna-X, fila-Y, entero-K.—Almacena la posición del mando de juegos (X, Y), y un valor entrado desde teclado.

POKE dirección, entero-N.—Almacena el valor N especificado (0-255) en la posición de memoria indicada.

NOTAS DE TRADUCCION

POLL dispositivo-1, dispositivo-2, ...—Envía una señal (POLL) a los dispositivos periféricos y devuelve el número de fichero y el estado del primer dispositivo que solicita una acción.

POP.—Borra la dirección de retorno almacenada por la última sentencia GOSUB de manera que RETURN retornará a la dirección almacenada por un GOSUB u otra sentencia similar anterior.

POS (expresión numérica).—Devuelve la posición actual del cursor.

POS cadena-A, cadena-B, entero-M.—Devuelve la posición de B (entero) en A, opcionalmente comenzando desde el carácter M de A.

POSITION fila-X, columna-Y.—Coloca el cursor en la posición especificada (X, Y) de la pantalla de gráficos.

POSITION nombre de fichero, entero-R.—Abre el fichero especificado (si no lo está ya) y mueve el puntero el número de campos especificado a partir de la posición actual.

POSITION número de fichero CLAVE cadena-A.—Posiciona el puntero de registro de un fichero indexado en la clave especificada.

PPOINT (fila-X, columna-Y).—Comprueba la posición (X, Y) de la pantalla de gráficos y devuelve 0 si OFF y el código de color si ON.

entero PRECISION entero.—Utilizado con PRINT; establece el número máximo de dígitos enteros y decimales que se pueden imprimir como una sola expresión.

NOTAS DE TRADUCCION

PRESERVE.—Utilizado con OPEN; especifica que se va a utilizar el área del fichero existente.

PRESET (fila-X, columna-Y).—Pone en ON la posición de la pantalla de gráficos especificada utilizando el color de fondo actual.

PR número de slot.—Selecciona un slot para recibir la salida de las sentencias PRINT subsiguientes.

PRI.—Ver PRINT.

PRINT lisa de impresión.—Provoca la impresión de los elementos de la lista especificada en la impresora estándar.

PRINT AT/@ entero-N, cadena de caracteres.—La impresión comienza en la posición de pantalla indicada por el entero.

PRINT LABEL expresión tipo cadena.—Escribe una etiqueta en el dispositivo de almacenamiento especificado.

PRINT USING cadena con formato de imagen.—Permite la salida de la impresión de acuerdo a un formato exacto determinado por la cadena de imagen (supresión de ceros y similares).

PRINTER IS (dirección de dispositivo).—Especifica el número de dispositivo y la anchura de página para todas las sentencias PRINT.

PRO.—Ver PROC.

NOTAS DE TRADUCCION

PROC.—Precede al nombre de un PROCEDIMIENTO (PROCEDURE) o función.

PROD$ (variable-A, variable-B, entero-P).—Calcula el producto de A por B redondeado a P posiciones decimales.

PROTECT nombre de fichero, código de protección/palabra clave (password).—Asigna un código de protección/password al fichero especificado para protegerlo de accesos no autorizados.

PROUND expresión numérica, posición de la potencia de diez.—Devuelve la expresión numérica redondeada a la posición de la potencia de diez indicada.

PSET (fila-X, columna-Y, código de color-N).—Pone en ON la posición de la pantalla de gráficos indicada utilizando el color de primer plano actual o el color N (opcional).

PT.—Ver PTR.

PTR número de canal/nombre de fichero/expresión numérica.—Mueve el puntero en un fichero secuencial para indicar cuál será el siguiente registro que se leerá/escribirá.

PTRIG.—Comprueba el botón del mando de juegos y devuelve 0 si está pulsado y 1 en caso contrario.

PURGE nombre de fichero.—Borra el fichero especificado del dispositivo de almacenamiento.

NOTAS DE TRADUCCION

PUT número de buffer, número de registro.—Escribe un registro desde el buffer del fichero especificado al disco (en forma directa), utilizando el número de registro actual o el especificado.

PUT nombre de fichero.—Escribe un registro en el dispositivo periférico.

PUT (fila-X, columna-Y) – (fila-A, columna-B), array-P, PSET/PRESET.—Visualiza el contenido del array-P en el rectángulo dado por la posición X, Y (esquina superior izquierda) y A, B (esquina inferior derecha) utilizando los colores de primer plano (PSET) o de fondo (PRESET).

NOTAS DE TRADUCCION

QUIT.—Cierra y salva automáticamente todos los ficheros antes de terminar el programa actual.

QUO$ (variable-A, variable-B, entero-P).—Calcula A/B redondeado a P posiciones decimales.

NOTAS DE TRADUCCION

R.—Ver RESET o RETURN o RUN.

RAD.—Establece la modalidad de RADIANES para las funciones trigonométricas.

RAD (entero).—Convierte de grados a radianes.

RAD$ (entero).—Convierte un número en una cadena de tres caracteres.

RADIAN.—Ver RAD.

RAN.—Ver RANDOMIZE.

RAMDOM.—Ver RANDOMIZE.

RANDOMIZE (expresión numérica).—Re-inicializa la semilla u origen del generador de números aleatorios.

RBYTE número de fichero, entero-N, variable-A.—Lee N bytes del fichero especificado y lo almacena en A.

RDRAW (columna-A, fila-B).—Dibuja una línea desde la posición actual (X, Y) a la posición relativa (X + A, Y + B).

REA (expresión numérica).—Devuelve la parte real de la expresión numérica especificada.

NOTAS DE TRADUCCION

REA. .—Ver READ.

READ variable-1, variable-2, ...—Lee las variables y les asigna valores a partir de las sentencias DATA correspondientes.

READ FILE nombre de fichero.—Lee los datos en formato binario desde un fichero al que se puede acceder en forma directa o secuencial.

READ LABEL variable tipo cadena/variable tipo array.—Devuelve en una variable tipo cadena la etiqueta del volumen de almacenamiento masivo que se esté utilizando, o las etiquetas de todos los volúmenes utilizados en ese momento, en una variable tipo array.

REAL variable numérica/array.—Dimensiona y reserva espacio para almacenar variables tipo cadena o array declarándolas como números REALES.

REC nombre/número de fichero.—Devuelve la posición actual del puntero de registro para el fichero de almacenamiento masivo especificado; si se indica un número negativo, devuelve la dirección física del registro.

RECALL nombre de fichero.—Utilizado para cargar en memoria un fichero/array previamente salvado en cinta de casete con STORE.

RECORDSIZE.—Especifica el número de bytes de un registro de datos.

RECOUNT.—Devuelve el tamaño del registro como el número de bytes del último registro de entrada.

NOTAS DE TRADUCCION

REDIM array (A, B)/cadena (A).—Define un nuevo tamaño de trabajo para un array; el número total de elementos no puede ser mayor que el declarado en la sentencia DIM original.

RELATIVE.—Ver ORGANIZATION.

RELEASE acceso a dispositivo.—Cancela cualquier SOLICITUD (REQUEST) de uso exclusivo de un dispositivo periférico, hecha por el programa o por la consola.

REM "cualquier combinación de caracteres".—Permite la inserción de comentarios no-ejecutables en el listado de un programa para proporcionar documentación y facilitar el seguimiento de la lógica.

REM (expresión numérica-X, expresión numérica-Y).—Devuelve el resto entero de la expresión X/Y.

REMARK.—Ver REM.

REN.—Ver RENUMBER.

RENAME nombre de fichero, nombre de fichero.—Ver NAME.

RENAME nombre de fichero TO nombre de fichero/código secreto.—Ver NAME.

RENUM.—Ver RENUMBER.

RENUMBER entero.—Renumera las líneas de un programa a partir de un número de comienzo (opcional) o a partir de 10 incrementándose de 10 en 10.

NOTAS DE TRADUCCION

Los números de líneas de programa referenciados en las sentencias tipo GOTO, ELSE también se renumeran adecuadamente para poder mantener la lógica de las bifurcaciones.

RENUMBER AT número de línea-1, incremento.—Renumera el programa cargado en memoria empezando opcionalmente en el número de línea y con el incremento especificados.

REP.—Ver REPEAT

REP$ cadena-A, cadena-B, desde-N, durante-M.—Sustituye M bytes de B en A a partir de la posición N.

REPEAT sentencia UNTIL condición.—Repite las sentencias del bucle hasta que se satisfaga la condición.

REPEAT$ (cadena-A, entero-N).—Devuelve una cadena en la que se repite la cadena-A un número N de veces.

REPLACE$ (cadena-A, cadena-B, cadena-C, entero-N, entero-M).—Devuelve la cadena-A en la que se han sustituido N ocurrencias de B por la cadena-C empezando en la posición M de la cadena-A.

REPO.—Ver REPORT.

REPORT.—Imprime la última condición de error del programa.

REQUEST dirección de dispositivo/variable de espera.—Reserva para la consola el uso exclusivo del dispositivo periférico especificado; se cancela mediante RELEASE.

NOTAS DE TRADUCCION

RES.–Devuelve el resultado de la última expresión ejecutada desde el teclado.

RES.–Ver RESTORE.

RES..–Ver RESTORE.

RESAVE nombre de programa, número de línea-1, número de línea-2.–Vuelve a salvar un programa completo o un bloque de líneas de programa en un fichero previamente creado con SAVE.

RESET (columna-X, fila-Y).–Pone en OFF la posición especificada (X, Y) de la pantalla de gráficos.

RESET nombre de fichero.

REST.–Ver RESTORE.

RESTORE número de línea.–Reposiciona el puntero de la sentencia DATA al comienzo de la DATA especificada (o en la DATA con numeración más baja, si no se especifica nada), de manera que se pueden volver a utilizar los valores. Si la línea especificada no corresponde a una sentencia DATA, el puntero se posiciona en la primera DATA que haya a continuación de esa línea.

RETRY.–Vuelve a intentar la ejecución de una sentencia que ha causado un error de ejecución de programa.

RESUME número de línea.–Termina la rutina de investigación de error (ON ERROR) y retorna incondicionalmente a la línea de programa especificada.

NOTAS DE TRADUCCION

RETURN variable.—Es la última línea de una subrutina y devuelve el control a la línea que hay a continuación de la sentencia GOSUB. RETURN se utiliza también con DEFFN para especificar un valor de la subrutina que hay que devolver al programa llamador.

REVISION.—Devuelve el valor que indica el nivel de revisión del sistema operativo asignado al software por el fabricante.

RI.—Ver RIGHT.

RIGHT (cadena-A, entero-N).—Devuelve los N caracteres más a la derecha de la variable tipo cadena especificada.

RIGHT$.—Ver RIGHT.

RMOVE columna-A, fila-B.—Mueve el cursor de gráficos sin dibujar nada desde la posición actual (X, Y) a la posición relativa (X + A, Y + B).

RND (entero).—Genera un número decimal (pseudo) al azar entre 0 y 1 (ó entre −32768 y 32767).

ROM.

ROT = entero-R.—Usado con DRAW y XDRAW para girar figuras de 0 a 360 grados dependiendo del valor de R (0-64).

ROTATE ángulo.—Usado con DRAW y RMOVE para producir un movimiento angular del cursor.

NOTAS DE TRADUCCION

ROW (array).—Devuelve el número de filas del array especificado.

**RPT$.**—Ver REPEAT$.

RSET variable tipo cadena = "juego de caracteres".—Almacena datos, justificados a la derecha, en la cadena especificada (normalmente en el buffer de entrada/salida).

RU.—Ver RUN.

RUN nombre de programa/nombre de fichero, número de línea.—Empieza la ejecución del programa cargado en memoria en la línea especificada (o en la que tenga la numeración más baja). Si se especifica un nombre de fichero, el programa se carga automáticamente desde el fichero y después RUN.

RUNNH.—Como RUN pero sin líneas de cabecera (NO HEADING) para el nombre de programa y la fecha, en el programa.

NOTAS DE TRADUCCION

S.–Ver SET o STEP o STOP.

SA.–Ver SAVE.

SAVE nombre de programa, número de línea-1, número de línea-2.–Lista y almacena todo el programa (o algunos bloques de líneas) cargado en memoria en un fichero de datos.

SCALARS.–Ver CLEAR CHARACTER/NUMERIC SCALARS.

SCALE entero.–Establece el factor de escala para las visualizaciones de gráficos.

SCR.–Ver SCRATCH.

SCRATCH carácter de control.–Borra el programa actual y restaura las variables a cero o a nulos; dispone de los parámetros:

A borra toda la memoria.
C restaura todas las variables incluyendo COMMON.
P borra el programa, las rutinas binarias y restaura variables.
V restaura todas las variables excluyendo COMMON.
– todas las teclas de función, o las especificadas, se borran con KEY.

SCRATCH número de fichero.–Borra todos los datos del canal/fichero especificado.

SCREEN modalidad, código de color.–Establece la modalidad de la pantalla (0 = TEXTO, 1 = GRAFICOS) y especifica el código de color para seleccionar los colores a partir de una lista codificada.

NOTAS DE TRADUCCION

SCRN (columna-X, fila-Y).—Devuelve el código de color de la posición (X, Y) especificada.

SEC.—Devuelve el número de segundos que han transcurrido desde la medianoche.

SECRET nombre de fichero, palabra secreta.—Establece una protección a través de una palabra secreta (password) para un fichero de manera que no pueda ser accedido desde otros programas.

SECURE número de línea-1, número de línea-2.—Evita que una línea, o un grupo de líneas, pueda ser LISTada; en el listado aparece un asterisco después del número de línea, sustituyendo a la línea del listado.

SEG$ (cadena-A, entero-N, entero-M).—Extrae de la cadena especificada, a partir del carácter N, un número M de caracteres.

SEQUENTIAL.—Ver ORGANIZATION.

SET (fila-X, columna-Y).—Pone en ON la posición X, Y de la pantalla de gráficos.

SET COLOUR.—Ver SETCOLOUR.

SETCOLOUR registro, código de color, brillo.—Almacena las combinaciones de color (0-15) y brillo (0-14) en los registros (0-4) a los que se accederá más tarde con COLOUR.

SETDOT.—Ver SET.

NOTAS DE TRADUCCION

SGET nombre/número de fichero, cadena-A.—Lee datos de un fichero random y los almacena en la cadena especificada.

SGN (expresión numérica).—Devuelve +1 si la expresión evaluada es positiva, −1 si es negativa y posiblemente 0 si la expresión es = 0.

SGN$ (expresión numérica).—Devuelve el valor "+" si la expresión evaluada es positiva, "−" si es negativa y "espacio" si es igual a cero.

SHFT (expresión numérica, entero-N).—Desplaza N bits a la izquierda (N positivo) o a la derecha (N negativo).

SHLOAD.—Carga desde cinta de casete una tabla de gráficos en alta resolución.

SHORT variable numérica/variable tipo array.—Dimensiona y reserva espacio para variables simples o de tipo array y las declara como de precisión SHORT.

SHOW variable-1, variable-2, ...—Utilizado con BREAK; visualiza los valores de las variables especificadas del programa que se está ejecutando.

SHUT nombre/número de fichero.—Cierra un fichero de manera que ya no puede ser accedido.

SIN (expresión numérica).—Devuelve el valor del seno del ángulo representado por la expresión numérica.

SINGLE.—Ver E.

NOTAS DE TRADUCCION

SINH.—Ver SNH.

SINPUT cadena-A, cadena-B, .../array-A, array-B, ...—Introduce líneas completas de texto en las cadenas y/o arrays especificados.

SIZE.—Devuelve el tamaño en bytes del programa cargado en memoria y la cantidad que queda disponible.

SIZE número de fichero.—Devuelve el tamaño del fichero especificado, en registros lógicos; si se especifica un número de fichero negativo se obtiene el tamaño del registro lógico en palabras.

SKIPF "nombre de fichero".—Salta al final del fichero (en cinta de casete) especificado o al final de la cinta.

SLEEP (entero-N).—Suspende la ejecución del programa durante N segundos.

SLEN número de fichero.—Devuelve el número de caracteres de tipo cadena que hay en el fichero empezando en la posición actual del puntero de palabra. Si no existen datos de tipo cadena se devuelve −1.

SO.—Ver SOUND.

SORT.—Mandato de utilidad del DOS para ordenar los nombres de los ficheros, en secuencia.

SOUND V, T, D, S/T, I.—Genera una nota específica a partir de los parámetros VOZ, TONO, DISTORSION y SONORIDAD; genera una nota con el TONO especificado y durante un determinado INTERVALO.

NOTAS DE TRADUCCION

SPA expresión numérica-N.—Define una cadena de N caracteres. Utilizado con PRINT y DISP para enviar un número determinado de espacios blancos hasta el final de la línea actual; N es una expresión numérica positiva redondeada a entero.

SPACE.—Ver SPACE$.

SPACE$ (entero-N).—Define una cadena de N espacios.

SPC.—Ver SPACE$.

SPEED expresión numérica.—Cambia el régimen de velocidad de salida/visualización de los caracteres; varía de 0 (el más lento) a 255.

SPOS FILE (fichero), expresión numérica.—Mueve el puntero de fichero a la positivo del byte del registro, indicada por la expresión numérica evaluada.

SPRINT cadena-A, cadena-B, ...—Produce la salida continua de caracteres tipo cadena sin caracteres de control de nueva línea e ignorando los márgenes establecidos.

SPR$ (cadena-A, cadena-B, cadena-C, entero-N).—Devuelve una cadena con todas (opcionalmente las N) ocurrencias no superpuestas de B en A, sustituidas por C.

SPUT nombre/número de fichero, cadena.—Escribe la cadena especificada en un fichero al azar.

SQR (expresión numérica).—Devuelve la raíz cuadrada de una expresión no negativa.

NOTAS DE TRADUCCION

SQRT.—Ver SQR.

SST.—Ver MID.

ST.—Ver STEP.

ST.—Ver STOP o STATUS.

STANDARD.—Establece el modo estándar para la salida de valores numéricos.

STATUS número de fichero, variable-A.—Comprueba la situación del fichero/dispositivo especificado y devuelve el código de condición en A (utilizado para la verificación después de las operaciones de lectura y escritura).

STE.—Ver STEP.

STEP entero-N.—Interviene en el FOR-NEXT para incrementar el contador de bucle en N para cada ciclo.

STEP número de línea.—Ejecuta el programa actual línea a línea empezando en la línea especificada.

STICK.—Devuelve un código dependiendo de la dirección en que se ha pulsado el mando de juego (se permiten 9 códigos).

STMA tipo, lista de parámetros.—Mandato de múltiples funciones dependiendo del valor del tipo, usado para la conversión ASCII/EBCDIC, restauración de bucles FOR/NEXT y GOSUB/RETURN y similares.

NOTAS DE TRADUCCION

STO.—Ver STOP.

STOP.—Termina la ejecución del programa actual y pone el puntero de programa en la línea de programa con numeración más baja.

STORE nombre/número de fichero.—Almacena todas las líneas de programa y rutinas binarias en un fichero de programas del dispositivo de almacenamiento especificado.

STORE BIN nombre/número de fichero.—Almacena todos los programas binarios de usuario cargados en memoria en el dispositivo de almacenamiento especificado.

STORE KEY nombre/número de fichero.—Almacena todas las definiciones para ayuda de las teclas especiales de función en un fichero especial (key file).

STR.—Ver STRING$.

STR$ expresión numérica-N.—Convierte el valor numérico de la expresión en una cadena.

STR.—Ver STR$.

STRI.—Ver STRING$.

STRIG.—Comprueba la posición del mando de juego y devuelve 0 si se ha movido y 1 en caso contrario.

NOTAS DE TRADUCCION

STRING.—Ver STRING$.

STRING$ (entero-N), cadena-A.—Produce una cadena con N repeticiones de A.

STUFF.—Ver POKE.

SUB nombre de subprograma, lista de parámetros.—SUB es la primera sentencia de un subprograma que será llamado con CALL por otro programa; la lista de parámetros contiene los nombres de funciones y elementos de datos a los que se le darán valores con el mandato CALL.

SUB$ (cadena-A, entero-N, entero-M).—Devuelve el carácter N-simo de la cadena especificada; opcionalmente puede devolver M caracteres empezando en el N.

SUBEND.—Ultima línea de una rutina/subprograma. Provoca un retorno al programa llamador.

SUBEXIT.—Provoca la terminación inmediata de la rutina y devuelve control a la primera sentencia que sigue al CALL desde el que se llamó al subprograma.

SUM.—Ver SUM$.

SUMMON nombre de programa.—Llama a una rutina en lenguaje ensamblador.

SUM$ (expresión numérica-A, expresión numérica-B).—Devuelve el valor de A + B.

NOTAS DE TRADUCCION

SWAP nombre de programa THEN GOTO número de línea/CON.—Carga y ejecuta un programa BASIC almacenado con SAVE y comienza la ejecución en la línea especificada o continúa (CON) la ejecución desde el punto en que se paró la última vez; al terminal devuelve control al programa actual.

SWAP% (array-A, array-B)/(variable-A, variable-B).—Utilizado para intercambiar los valores almacenados en dos arrays o variables definidos de igual forma.

SWAP% (entero-N).—Efectúa una operación de intercambio con los dos bytes del entero-N especificado.

SYS.—Ver SYSTEM.

SYS (entero).—Devuelve las constantes del sistema (fecha, hora, etc.) dependiendo del valor del entero.

SYSTEM.—Permite la lectura de programas en lenguaje máquina (código objeto) desde casete a memoria.

NOTAS DE TRADUCCION

T.—Ver TAB o THEN o TAN.

TAB (posición de impresión).—Utilizado con PRINT, DISP y LDISP. El siguiente item se sacará en la posición indicada, rellenando a blancos el espacio existente entre la posición actual y la especificada.

TAN (expresión numérica).—Devuelve la tangente del ángulo dado por la expresión.

TAND.—Ver TAN.

TANG.—Ver TAN.

TANH.—Ver TNH.

TAPPEND.—Añade, desde la cinta del casete, el programa especificado, al que hay cargado en memoria.

TEXT.—Cambia la modalidad de pantalla de GRAFICOS a TEXTO.

TH.—Ver THEN.

THE.—Ver THEN.

THEN.—Ver IF.

TI.—Ver TIM.

NOTAS DE TRADUCCION

**TI$.**—Ver TIME$.

TIM entero.—Devuelve el tiempo de ejecución del programa actual expresado en segundos, los minutos, la hora, el día y el año actual dependiendo del valor del entero especificado.

TIME.—Ver TIM.

TIME.—Devuelve el tiempo transcurrido en segundos desde la medianoche anterior.

TIME$.—Devuelve la hora del día en formato de 24 horas (HH MM SS).

TIME$ (entero-N).—Devuelve el tiempo desde la medianoche como N minutos; si N es cero devuelve la hora actual del día.

TIMER.—Mide el tiempo en 1/60 de segundo.

TINPUT entero-N.—Como INPUT pero para una duración de Nx1/10 segundos.

TLIST número de fichero.—Lista el tamaño y el nombre de todos los ficheros contenidos en el dispositivo/número de fichero especificado.

TLOAD nombre de programa.—Carga en memoria desde casete el programa especificado.

TNH entero.—Calcula la tangente hiperbólica del número especificado.

NOTAS DE TRADUCCION

TOP.—Identifica la dirección del primer byte de la memoria no utilizada (es decir, el siguiente byte después del programa actual.

TR.—Ver TRACE.

TRACE número de línea-1, número de línea-2.—Sigue la TRAZA de la lógica de todo o parte del programa en su ejecución; cualquier bifurcación produce una salida en la impresora del sistema que identifica el origen y el destino de la bifurcación; los números de línea desde/hasta delimitan el área del programa para la cual se activa la TRAZA.

TRACE ALL.—Sigue la TRAZA de la lógica del programa y las asignaciones de variables; es similar a ejecutar las TRACE y TRACE ALL VARIABLES.

TRACE ALL VARIABLES número de línea-1, número de línea-2.—Monitoriza e imprime los cambios en los valores de todas las variables, en todo o en un segmento del programa.

TRACE OFF.—Ver TROFF.

TRACE ON.—Ver TRACE.

TRACE PAUSE número de línea, expresión numérica.—Utilizado como "breakpoint" (puntos de control). Provoca un alto en la ejecución del programa antes de que una línea determinada sea ejecutada un número opcional de veces.

TRACE VARIABLES lista de variables.—Como TRACE ALL VARIABLES para los arrays/variables especificados.

NOTAS DE TRADUCCION

TRACE WAIT entero-N.—Utilizado con cualquier sentencia TRACE selectiva o TRACE ALL. Hace que el programa espere un número determinado de milisegundos después de cada ejecución de una línea de programa que dé lugar a una salida de impresión.

TRAP número de línea.—Bifurca al número de línea especificado si se produce un error de ejecución de programa; después de su utilización hay que restaurarla. Con PEEK se puede obtener el código de error almacenado y el número de línea del programa en error.

TRIM$ expresión tipo cadena.—Devuelve la expresión tipo cadena en la que se han borrado los blancos de cabecera y cola.

TRM$ (cadena-A).—Devuelve la cadena-A con los blancos de cola borrados.

TRN.—Ver MAT.

TROFF.—Desactiva el TRACE.

TRON.—Activa el TRACE.

TRUE.—Devuelve un valor −1 si se satisface la condición BOOLEAN TRUE (VERDADERO).

TRUN$ (cadena-A, 1).—TRUNCA la cadena especificada en el primer nulo, carácter de retorno de carro o de salto de página si no se especifica 1. Si se especifica 1 trunca la cadena en el primer carácter distinto de espacio.

TSAVE.—Almacena el programa actual en cinta de casete.

NOTAS DE TRADUCCION

TYP nombre/número de fichero.–Devuelve un valor indicando si el fichero pecificado está disponible o no. Si lo está indica el TIPO (TYPE) de da a los que se accederá a continuación.

TYPE.–Ver TYP.

NOTAS DE TRADUCCION

U.–Ver UNTIL.

UCALL subrutina.–Llama y ejecuta una rutina en lenguaje ensamblador.

UCM$ (cadena-A).–Inversa de la operación CRAM. Desempaqueta cada 2 bytes en tres caracteres (A-Z, 0-9, .– espacio y coma).

UNBREAK (número de línea-1, número de línea-2).–Utilizado para cancelar todos los puntos BREAK; si el programa se ha parado es necesario RESUME para continuar la ejecución.

UNLESS condición.–Permite la ejecución de una sentencia solamente si la condición especificada es falsa.

UNLOCK número de fichero.–Desbloquea un fichero de datos previamente bloqueado o hecho inaccesible (LOCK) para otros programas.

UNPLOT columna-X, fila-Y.–Pone en OFF la posición de pantalla especificada.

UNTIL.–Ver IF, THEN, ELSE, WHILE.

UNTRACE (variable-1, variable-2, ...).–Detiene la función TRACE por completo o sólo para las variables especificadas.

UPC.–Ver UPS$.

NOTAS DE TRADUCCION

**UPC$.**–Ver UPS$.

UPDATE número de fichero.–Vuelve a escribir en un fichero indexado el registro leído con el GET más reciente.

UPS$ (expresión tipo cadena).–Devuelve la cadena especificada con los caracteres de minúsculas convertidos a caracteres de mayúsculas.

USER.–Ver USR.

USER$.–Devuelve en una cadena de caracteres las identificaciones de los usuarios actuales.

USING.–Ver PRINT.

USER n, expresión numérica.–Da control a una rutina en lenguaje máquina previamente definida por DEFUSRn (n = 0 a 9); la expresión numérica evaluada es un argumento entero.

USR nombre de subrutina.–Da control a una subrutina en lenguaje máquina.

USRID.–Devuelve el número de identificación del usuario o el número asignado a la consola activa en ese momento.

NOTAS DE TRADUCCION

VAL expresión tipo cadena.—Devuelve el valor numérico de la cadena de dígitos especificada, de manera que se pueda utilizar su valor para cálculos posteriores.

VALS expresión numérica.—Devuelve la cadena que representa la expresión numérica especificada que puede utilizarse en impresión.

VARIABLE.—Ver ORGANIZATION.

VARPTR variable-V.—Devuelve la dirección y el valor de la variable-V especificada.

VDU modalidad, lista de parámetros.—Control en modalidad de gráficos para localización del texto, cambio de color y cambio del origen de los gráficos dependiendo de la modalidad establecida: 5, 19 y 29 respectivamente.

VERIFY nombre de fichero, número de disco, número de slot, número de volumen.—Comprueba la consistencia del fichero especificado.

VIEWPOINT entero-A, entero-B.—Establece el tamaño del área a visualizar en la pantalla de gráficos.

VIRTUAL.

VLIN fila-X, columna-Y AT columna-A.—Visualiza en la pantalla de gráficos una línea vertical desde la fila-X a la fila-Y empezando en la columna-A.

VP.—Ver VPOS.

NOTAS DE TRADUCCION

VPOS.–Devuelve la posición vertical del cursor, es decir, el número de columna.

VTAB fila-X.–Posiciona el cursor en la fila-X de la pantalla de gráficos para empezar la impresión.

NOTAS DE TRADUCCION

W.–Ver WIDTH.

WAIT (número de milisegundos).–Retarda la ejecución del programa actual el número de milisegundos especificado.

WBYTE número de fichero, variable-A, entero-N.–Escribe en el fichero especificado N caracteres de la variable-B.

WEAVE nombre de programa.–Fusiona el programa especificado con el programa actual entrelazando las sentencias según sus números de línea.

WEND.–Ultima sentencia de un bucle WHILE.

WHATS "nombre de fichero".–Imprime la información de los atributos del fichero especificado.

WHILE condición.–Primera sentencia de un grupo que se va a repetir en un bucle hasta que la condición ya no se satisfaga.

WHO cadena.–Devuelve la identificación del usuario actual del terminal.

WINDOW mínimo, máximo.–Establece los límites MINIMO y MAXIMO de los valores a incluir en una visualización.

WRD (número de fichero).–Devuelve la posición del apuntador de palabra del fichero especificado (en almacenamiento masivo).

NOTAS DE TRADUCCION

WRITE.—Escribe datos en un fichero.

WRITE FILE.—Ver WRITE.

NOTAS DE TRADUCCION

XDRAW número de lista AT columna-X, fila-Y.—Borra una figura (desde una lista almacenada en memoria) que estaba dibujada en la pantalla de gráficos empezando en la posición X, Y y si no hay ninguna figura en este punto aparecerá una, hasta que el color de cada una de las posiciones de la pantalla de gráficos sea complementado o desactivado.

XLATE (cadena-A, cadena-C).—Convierte la cadena-A del código de almacenamiento interno actual al especificado por C (por ejemplo de ASCII a EBCDIC).

XOPEN número de canal, nombre de fichero.—Abre un fichero en disco para entrada y/o salida.

XOR.—O exclusiva en las sentencias IF.

XPOS.—Devuelve la posición del cursor sobre el eje X actual.

NOTAS DE TRADUCCION

YEA.

YPOS.–Devuelve la posición del cursor sobre el eje Y actual.

NOTAS DE TRADUCCION

ZER.—Ver MAT.

NOTAS DE TRADUCCION

DICCIONARIO DEL BASIC

Este Diccionario le ayudará a utilizar programas de un ordenador en otro. Contiene más de ochocientas entradas que suponen las acciones de casi todas las sentencias, comandos o funciones que Vd. puede encontrar o necesitar. Así, cuando en un programa encuentre una sentencia nueva, podrá comprender qué hacer y de qué forma hacerla funcionar en su propio ordenador. Cada entrada está explicada en un lenguaje sencillo.

Se incluyen en otras secciones:

- Un área de notas
- Una extensa descripción de la terminología BASIC
- Una aproximación sistemática a la traducción de programas BASIC
- Comparaciones entre PEEK, POKE y mapas de memoria de los principales ordenadores personales

En resumen, un libro de consulta continua que le ahorrará mucho tiempo y facilitará su trabajo, haciéndolo cómodo y eficaz.

EDITORIAL PARANINFO S.A.
Magallanes, 25 - 28015 Madrid
ISBN: 84-283-1378-4